CHARACTERIZATION OF THE LIPID PROFILE IN *Plasmodium vivax*

HUGO CARLOS SÁMANO SÁNCHEZ

(Bachelor on Genomic Sciences from the National Autonomous University of Mexico)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER IN INFECTIOUS DISEASES, DRUG DISCOVERY AND VACCINOLOGY

DEPARTMENT OF MICROBIOLOGY

NATIONAL UNIVERSITY OF SINGAPORE

AND

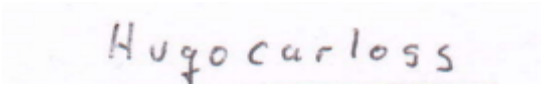
UNIVERSITY OF BASEL

2014

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Hugo Carlos Sámano Sánchez

December 24th 2014

Acknowledgements

It would be impossible to finish this master thesis without the help of several people. First and foremost I gratefully acknowledge the continuous academic advice and support of Dr. Bruce Russell but also his extraordinaire example as a group leader.

I would like to thank all the members of the Vivax Malaria Lab: Benoit, Jee Sun, Li Hua, May Ling, Poy and Rosemary for their help, kind company and friendship.

Special thanks to Dr. Markus Wenk for being my cosupervisor as well as Federico and Jacklyn, from the same group, for teaching a bioinformatician the lipidomics field.

Claudia, Collins, Esteban, Iris, Maggie and Manu: it has been a pleasure to share this time, back as students, in Basel and then in Singapore with you people.

I wish to thank the Swiss Tropical Institute, the University of Basel, the National University of Singapore and the Novartis Institute for Tropical Diseases for making possible this joint Master program.

Finally, to my family for their support through all this time.

Table of Contents

Summary

The emerging lipidomics field can be used to understand key differences in the phospholipid composition of human pathogens such as the malaria-causing parasites of the genus *Plasmodium*. While rapid advances in the technology used for high-throughput analyses have decreased the cost and time needed for lipidomics data generation, the interpretation of these large data sets has become a significant bottleneck. In this thesis we present the first lipidomics analysis of *Plasmodium vivax* (the most important cause of malaria in South America and Asia) infected Red Blood Cells. Importantly, we presented and described a package-structured code in R programming language used to rapidly process the *P. vivax* lipidomics data. This program acquires monoisotopic features from a Mass Spectrometry experiment to be processed in a subsequent Tandem MS experiment. It also identifies particular lipid species by working coupled to MSPepSearch and gives hints on lipid classification given a database of Precursor Ion and Neutral Loss Scans.

This novel pipeline was used to identify lipids being over expressed in reticulocytes infected with *P. vivax* in comparison with non-infected reticulocytes and the results are discussed in terms of an extensive literature review and the analysis of an additional lipid extraction from *P. falciparum*-infected cells.

List of Tables

List of Figures

Abbreviations

CB	Cord Blood
CDS	Phosphatidate Cytidylyltransferase
CE	Cholesteryl ester
Cer	Ceramide
DG	Diacylglicerol
ER	Endoplasmic Reticulum
ESI	Electrospray ionization
FAs	Fatty Acids
GPA	Glycerophosphatidic acid
GPCho	Glycerophosphatidylcholine
GPEtn	Glycerophosphatidylethanolamine
GPGro	Glycerophosphatidylglycerol
GPIns	Glycerophosphatidylinositol
GPSer	Glycerophosphatidylserine
IPC	Inositolphosphoceramide
iRBC	infected Red Blood Cell
LC	Liquid Chromatography
m/z	Mass-to-charge
MACS	Magnetic-Activated Cell Sorting
MG	Monoacylglycerol
MAQC	MicroArray Quality Control
mill	Millions
MS	Mass Spectrometry
MS/MS	Tandem Mass Spectrometry
NA	Not Applicable
Neg	Negative Ionization Mode
NLS	Neutral Loss Scan
NMR	Nuclear Magnetic Resonance
PA	Phosphatidic Acid
PC	Plasmenylphosphatidylcholine

PCA	Principal Component Analysis
PIS	Parent Ion Scan
Pos	Positive Ionization Mode
PVM	Parasitophorous Vacuole Membrane
QC	Quality Control
QTOF	Quadrupole Time-of-Flight
QTOF-MS	Quadrupole Time-of-Flight Mass Spectrometer
RBC	Red Blood Cells
SD	Standard Deviation
SM	Sphingomyelin
TG	Triacylglycerol
Tandem MS	Tandem Mass Spectrometry
TMS	Trimethylsilyl
UPLC/MS	Ultra-Performance Liquid Chromatography/MS
UPPv	More Abundant In the Infected Samples
UPRetic	More Abundant In Non-Infected Samples

Introduction: Part I, Lipidomics

The lipidomics field

The state-of-the-art of the research in science is characterised by the availability of technical approaches that allow a "to-down" interpretation of the biological systems. Together with the well-known fields of genomics, transcriptomics and metabolomics is the newer lipidomics discipline, which studies the large-scale identification of lipids and their interacting moieties (Wenk, 2005). The untargeted systems-level approach of lipidomics is an innovative and unbiased way to study the phospholipid composition of biological entities. The general idea of one of the current and most common methods for the description of such lipids is to use a Mass Spectrometer to identify the exact mass of a previously purified and ionized non-water-soluble metabolites followed by a second step of ionization after fragmentation that can lead to the identification of fragment ions that give structural information for a single lipid molecule. However, other experimental approaches can also be used like nuclear magnetic resonance (NMR) or Biochemistry assays, nonetheless, they allow only a low-throughput analysis.

The rapid improvement in lipid extraction protocols and the better signal-to-noise ratio in new mass spectrometers, have made possible to produce lipidomics studies using samples of limited quantity; for example in the study of the membrane composition of *Escherichia coli* (Oursel et al., 2007) or of plant lipids (Welti & Wang, 2004). Other cases are the comparative lipidomics, in which two or more related biological samples are compared such as the work of Singh et al. (Singh & Prasad, 2011) where they compared the lipid profiles of a sensitive and a resistant strain of the fungus *Candida albicans*, the identification of common and unusual polar lipids in the Apicomplexa parasite *Toxoplasma gondii* (Welti et al., 2007) or in mammalian organisms exploring

the effect of toxic compounds in the lipid composition of the liver by using NMR (Fernando, Bhopale, Kondraganti, Kaphalia, & Shakeel Ansari, 2011).

When a global search is performed using a single method in order to obtain a broad lipidomics coverage from a given sample, the study is called untargeted and comes with some advantages as well as some disadvantages. It can be used when the identity of the lipid target is unknown or several lipid classes are required to be analysed simultaneously with a quantitative output (Layre & Moody, 2013).

Basics of Mass Spectrometry

The only technique that can manage the analysis of thousands of different lipid molecules in a single run is Mass Spectrometry (MS). In principle, it does not require an initial separation of the sample but a direct infusion can be used (Li, Yang, Bai, & Liu, 2014). However, complex samples like tissues or body fluids can contain thousands of distinct lipid species (Layre & Moody, 2013). Coupling MS with a chromatographic separation can improve the identification by separating the inputs by lipid class or size.

For some authors the introduction of lipidomics started with the development of the Electrospray Ionization (Han & Gross, 2003; Li et al., 2014) where the idea is to apply an electric field to a liquid (the lipid extraction) in order to induce a charge, then, and after desolvation, charged lipids are separated by its mass-to-charge (*m/z*) ratio which is the ultimate goal of the Mass Spectrometer (Hoffman & Stroobant, 2007). When coupled to a liquid chromatography column, an additional dimension of information is generated for each component of the sample which is the retention time, this value depends on chemical properties like the length of the fatty acids, the volatility and polarity of the molecule and even the location of double bonds in the fatty acids (Dobson, Christie Ww Fau - Nikolova-Damyanova, & Nikolova-Damyanova).

For lipid identification purposes, molecules with a single charge are desired, as their *m/z* ratio will correspond to its mass. However, the number of charges acquired by a molecule will depend on how many ionisable sites has this molecule, which is a property that can correlate with its size. For example, cardiolipin, a phospholipid characteristic of the mitochondria, usually acquires a double charge (Han, Yang, Yang, Cheng, & Gross, 2006).

Mass spectrometers using a Quadrupole Time-of-Flight (QTOF) can positively or negatively ionize the sample giving rise to two sets of measurements, one corresponding to molecules that can get a positive charge, denoted by $[M + H]^+$ like in the case of phosphatidic acid (PA), phosphatidylglycerol (PG), phosphatidylinositol (PI), phosphatidylserine (PS), inositolphosphoceramide (IPC) and diacylglycerol (DG), or a negative charge as in the case of sterols, phosphatidylcholine (PC), triacylglycerol (TG), ceramide (Cer) and also DG (Ejsing et al., 2009), to mention some.

The mechanism used by a QTOF mass spectrometer (QTOF-MS) relies on accelerating the ions coming from a fragmentation of a 'precursor ion', they will have the same velocity as the precursor but different kinetic energies depending on their masses. Consequently, these ions will have different flight times (thus the name: time-of-flight). The resolution of such instruments will be proportional to the length of the flight path.

Although the most sensitive devices on the market have several systems to reduce false negatives (ion suppression), and obtain high accuracy on complex samples, the obtained data is the result of adducts, fragments and isotopic peaks that must be further analysed and interpreted (Tautenhahn, et al. 2008). Herein, the term "feature" will refer to a signal defined by a specific retention time and a mass-to-charge (*m/z*) ratio.

Isotopes

Natural molecules are composed of isotopes, which are atoms of the same chemical element with different masses. Considering the most common elements present in lipids, the lightest isotope is always the most abundant (Berglund & Wieser, 2011). Table 1 shows the most common heavy isotopes that can be found in lipid species. Sulfur has the biggest isotope variation, however it is rather restricted to particular lipids like cholesterol sulfates (e.g. in brain, kidney, and other mammalian tissues) or taurolipids (e.g. in the bile acids, in *Tetrahymena* and rarely in plants), between others (Ishizuka, 1997).

The isotopes ^{13}C, ^{33}S and ^{15}N are the next most abundant and they all contribute to "+1" to the mass of the molecule that contains them. The isotopes ^{18}O and ^{34}S, if present, will lead to "+2" to the mass of the lipid.

Table 1

Isotope	Mass [Da]	% Abundance
^{31}P	30.973763	100
^{1}H	1.007825	99.985
^{16}O	15.994915	99.76
^{14}N	14.003074	99.63
^{12}C	12.000000	98.90
^{32}S	31.972072	95.02
^{34}S	33.967868	4.21
^{13}C	13.003355	1.10
^{33}S	32.971459	0.75
^{15}N	15.000109	0.37
^{18}O	17.999159	0.20
^{17}O	16.999131	0.038
^{2}H	2.014102	0.015

Table 1. Natural abundance of isotopes commonly present in lipid species. Data ordered by abundance and obtained from the latest report from IUPAC (Berglund & Wieser, 2011).

To illustrate the effect of the isotopes to the distribution of the *m/z* values of a simple molecule, we will consider the alkane $C_{20}H_{42}$. The most abundant molecule, based on Table 1 will be the molecule composed of twenty ^{12}C atoms and forty-two ^{1}H atoms with a probability of $0.989^{20} \times 0.99985^{42}$ which is 0.7998757, then we can calculate the

mass of this molecule as 12.000x20+1.007825x42 which is 282.3286. Similarly, we can obtain the mass and probability for the second most likely alkane of 20 carbons, assuming the distribution values from Table 1. Table 2 summarizes just the first three theoretically most common molecules.

Table 2

Molecule	Exact mass	Abundance	Normalized Abundance
$^{12}C_{20}{}^{1}H_{42}$	282.3286	0.7999	0.8033
$^{12}C_{19}{}^{13}C_{1}{}^{1}H_{42}$	283.332	0.1772	0.1779
$^{12}C_{18}{}^{13}C_{2}{}^{1}H_{42}$	284.3354	0.0187	0.0188
Sum:		0.9958	1

Table 2. Example of the abundance of some isotopes of $C_{20}H_{42}$. The abundance was calculated using the values from (Berglund & Wieser, 2011). The normalized abundance is presented to explain the proportion of the peaks in a mass-to-abundance plot.

The mass of a molecule considering only the most abundant atoms is called *monoisotopic mass* and the *monoisotopic mass spectrum* refers to the spectrum that contains the molecules made up of the first most common isotopes. For small molecules like $C_{20}H_{42}$, the most likely, or abundant, mass will be the monoisotopic mass (see Table 1 and Figure 1). However, it is easy to note that for bigger molecules the probability to find a heavier isotope increases. Given that Mass Spectrometers are able to distinguish isotopes, the monoisotopic mass will be used hereafter in this thesis instead of any other possible definition of the mass of a molecule.

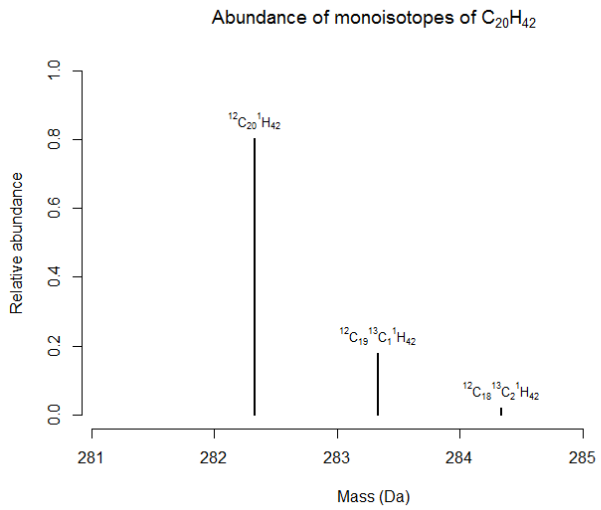


Figure 1. Relative abundance of the three most common isotopic masses of the alkane $C_{20}H_{42}$ based on the abundances reported by (Berglund & Wieser, 2011). An approximate difference of '+1' in the mass value is observed in the molecule containing one ^{13}C atom with a significant relative abundance of ~17.8%, which corresponds to ~17.7% in natural observations (see Table 1). The third isotopic peak and all other not-depicted heavier isotopes are very rare.

Adducts

Neutral molecules can get ionized not only with a proton but with another ion, when this occurs the product is called adduct. Examples of 'ionizing' ions are sodium, potassium, ammonium, chloride and acetate. The production of adducts with depend on the head group of the lipid, how carefully it was desalted or the solvent used for the lipid extraction (Hoffman & Stroobant, 2007).

Tandem Mass Spectrometry

Current Mass Spectrometer instruments allow the discrimination of masses with a mass accuracy under 1 ppm (as with the 6550 iFunnel QTOF-MS from Agilent) corresponding to +/- 0.0016 m/z for heavy lipids. However different molecules with the same atomic composition can have different chemical properties and structures. For example, the compounds Palmitoyl-EA, 3-ketosphinganine and Sphingosine have all the same composition: $C_{18}H_{37}NO_2$ but different structures (Fig 2.) and will produce a peak at the same m/z value. Tandem Mass Spectrometry (Tandem MS or MS/MS) is a

procedure where the fragmentation of a previously isolated monoisotopic ion (the precursor) is used to analyse in a mass spectrometer again the mass of its components. A precursor ion carrying only one charge can be fragmented by collision with an inert gas into a lighter ion and a neutral fragment. By using Tandem MS, molecules like Palmitoyl-EA and Sphingosine can be distinguished as fragmentation patterns will be different, even differences in stereochemistry can lead to different spectra (Hoffman & Stroobant, 2007).

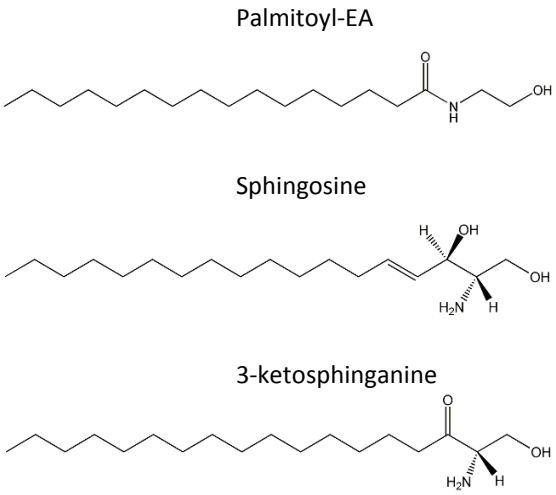


Figure 2. Structure of three molecules with the same atomic composition but different structure. From top to bottom: Palmitoyl-EA (LMFA08040013), Sphingosine (LMSP01010001) and 3-ketosphinganine (LMSP01020002). In parenthesis are the identifiers in Lipid MAPS database.

In further detail, the measurements obtained in an MS/MS experiment can be divided in, but not restricted to, 'product ion scan' or 'neutral loss scan'. The difference relies on which fragment is being determined, if the charged molecule or the complementary molecule (without charge).

To exemplify some of the rules governing the fragmentation patterns in a Tandem MS experiment, the synthetic ceramide 1,3-di-*O*-trimethylsilyl-*N*-stearoyl sphingosine will be used (Samuelsson & Samuelsson, 1960). For the sake of simplicity, rounded numbers (to integers) will be used, which is the way MS research was performed before high-resolution mass spectrometers were available. When the precursor ion, the intact

and pure molecule but positively charged, is submitted to electron impact, the following ions will be observed: one ion at *m/z*=694 (15 units at the left of the precursor ion, also represented as $[M\text{-}15]^+$ indicating the loss of 15 *m/z* units leading to a positive ion) corresponds to the precursor after losing a methyl group (CH_3, with mass 15 Da). Other molecules can be separated like trimethylsilanol (90 Da) or the terminal methylene (103 Da) giving rise to ions with *m/z* values of 619 and 606, respectively. The fatty acid with 18 Carbons labelled with the letter 'b' in Figure 3 can form a fragment with the molecular formula $C_{18}H_{36}N_1O_1$ (282 Da), however, and in order to generate a molecule with a positive charge (also called 'product ion') only molecules that have lost a neutral fragment will be detected (called 'neutral loss') giving rise to the retention of an additional Hydrogen $[M\text{-}(282+1)]^+$ with a peak at *m/z*=426. Additional combinations generated by the simultaneous loss of the 18-Carbons fatty acid and the trimethylsilanol ($[M\text{-}283\text{-}90]^+$) will create the peak at *m/z*=336. Another option is the cleavage between C-2 and C-3, creating the fragments labelled as 'a' (*M*-d; *m/*z=311) and 'd' (M-a; *m/*z=398), where the last one has the highest abundance making it the reference for the rest of the peaks. Finally, more complex fragmentations can occurs as in the case of the peak at *m/z*=471, where a cleavage between C-2 and C-3 was followed by a transfer of the trimethylsilyl (73 Da) group to the remaining molecule ($[M\text{-}(a\text{-}73)]^+$).

Additional details about the Mass Spectrometry technology go beyond the scope of this thesis.

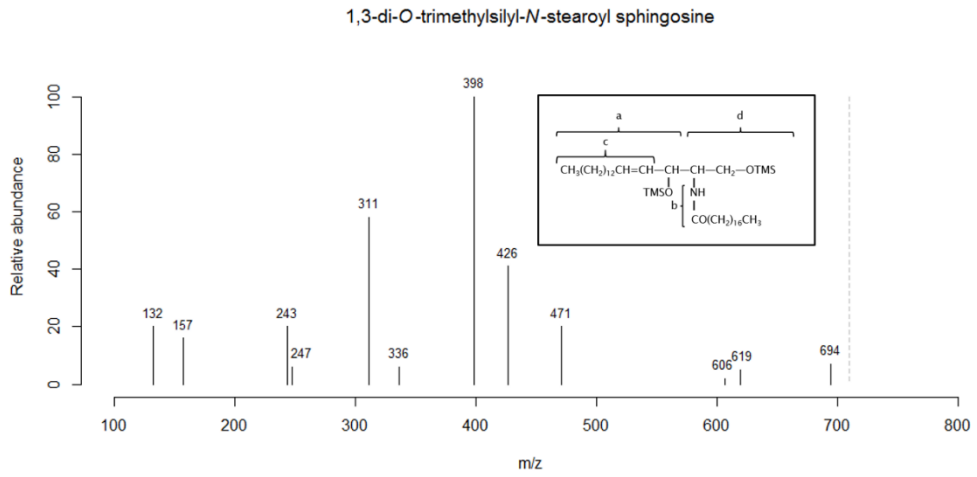


Figure 3. The fragmentation spectra of the synthetic ceramide 1,3-di-O-trimethylsilyl-N-stearoyl sphingosine (depicted at the m/z=709 with a gray dashed peak) based and simplified from the data published by (Samuelsson & Samuelsson, 1960). TMS, trimethylsilyl.

Processing of Mass Spectrometry Data

Mass Spectrometry experiments preceded by Liquid Chromatography (LC) can produce massive amounts of data (more than three thousand signals per experiment) that require a special preprocessing before well-behaved and reproducible peaks can be identified. Information on peak intensity and the mass-to-charge ratio can be used together with the retention time that is dependent on the used chromatographic column for data filtering and correction.

In order to compare more than one MS experiment, discreet and aligned peaks defined by *m/z* and intensity values are needed. Such alignment can be performed with XCMS (Smith, Want, O'Maille, Abagyan, & Siuzdak, 2006), a software written in R programming language for metabolomic analysis. It has become broadly used and a proof of its success is the publication of several improvements to either the feature detection algorithm (Tautenhahn, Bottcher, & Neumann, 2008) or to the online version (Gowda et al., 2014) which is user-friendly although less flexible as will be deeply addressed in the discussion.

Bioinformatics tools for structure identification of lipids

The process previously described for the identification of the structure based on Tandem MS data can be automatized to identify possible ions with a known *m/z* value or combinations of them. This will essentially depend on how big, specific and accurate the database of spectra is. As further discussed in the next section, a big progress is needed in different aspects of the lipidomics research and one of them is the development of databases fulfilling the mentioned characteristics. The best effort done so far, in quantitative terms, is LipidBlast (Kind et al., 2013). Published in 2013, this database contains computer-generated tandem mass spectra from 119,200 lipid compounds which surpasses almost ten times the total number of compounds from other libraries like Metlin, MassBank and NIST. It includes 26 lipid classes, including phospholipids, modelled under different observed adduct ions in, both, positive and negative ionization modes. The construction of the library is based on Lipid MAPS and associated tools (Fahy, Sud, Cotter, Cotter, & Subramaniam, 2007) but generously complemented with lipids found in chloroplasts, including mono and digalactosyldiacylglycerols as well as sulfoquinovosyldiacylglycerols (Benning, 2008). Given that ion abundance patterns can be instrument-specific, LipidBlast modelled spectra for different mass spectrometers, including QTOF-MSs.

The next step is the identification of the spectra of an unknown compound within a given library in a reasonable time and the assignment of a score that will allow the discrimination of a true positive. NIST MS Search Program (Stein & Scott, 1994), developed by the National Institute of Standards and Technology, performs such search based on a two-step algorithm. Briefly, the peaks of the query and the library are scaled by their abundance and *m/z* values. Then, and starting with the highest scaled peak in the query, one by one the peaks of the query are compared with a subset containing the most intense peaks in the scaled library. The way NIST MS Search solved the problem of scoring the hit results is by taking advantage of a multidimensional hyperspace,

where each mass spectrum will represent a single point described by a vector with the intensity values for each *m/z* variable. Thus, when a query and a target spectra are identical, both point representations will coincide, while two similar spectra will have point representations close to each other. The similarity can be measured with the dot-product of the vectors describing both points. An improved scoring system also considers the ratios of adjacent peaks. The mentioned algorithm is implemented in a Graphical User Interface for easy manual inspection (Download from: http://chemdata.nist.gov/) or in a high-throughput mode (Download from: http://peptide.nist.gov/software/ms_pep_search_gui/MSPepSearch.html).

Data analysis: The normalization in untargeted Mass Spectrometry analyses

As mentioned before, lipidomics is still an emerging field and not as developed as other system-level approaches. Thus, some methodologies for the processing of the data obtained on MS or MS/MS experiments are still under discussion. For example, there is some controversy regarding the use of normalization methods; below are discussed some opposing approaches to data normalization.

It has been proposed the median fold change normalization, which centers each distribution in the median of the fold changes, to reduce dilution-induced variation in metabolic profiles using ultra-performance liquid chromatography/MS (UPLC/MS) (Veselkov et al., 2011). The same procedure was used in a very recent study applying UPLC/MS to assess radiation-induced metabolic changes in human fibroblasts (Kwon et al., 2014).

Another group proposed quantile normalization to compare metabolomic data from different LC-MS studies (J. Lee et al., 2012), this method is the standard for the analysis

of microarrays for gene expression, where the distribution of the intensity values from the different samples are adjusted to be similar in statistical properties.

There are different, and sometimes inconsistent, ways to evaluate a normalization procedure. Some authors use a multiparametric method like Principal Component Analysis (PCA) to verify that replicates cluster together. Other options are to make boxplots of the intensity values per sample, quantile-quantile plots or by observing the standard deviation of signal intensity as a function of the Quality Control (QC) intensities ordered by ranks (Veselkov et al., 2011) (J. Lee et al., 2012).

It is important to note that in the transition from research to the clinical practice, microarray and high-throughput sequencing technologies needed an optimized framework for quality control. The project regulating this is called MAQC (for MicroArray Quality Control) for the case or microarrays and SEQC for next-generation sequencing (Shi et al., 2006) (DeLuca et al., 2012). Nothing similar has been published yet for lipidomics studies showing the importance in discussing the current tools and pipelines to process this kind of data.

The R programming language

R is a programming language for statistical computing that has been widely used to develop tools for the analysis of high-throughput biological data. It works through the addition of 'packages' which are sets of functions or subroutines that are developed by the scientific community. With the aim to centralize, quality assess users-developed R packages, a repository called Bioconductor was created in 2001, since then, tools required for the new technologies have been evaluated and distributed by this repository (Gentleman et al., 2004).

The package structure allows a high degree of flexibility during the processing of data, as several independent functions, either previously published or in-house developed,

can be concatenated. R is also an easy language for statistical analyses and the creation of graphics. Furthermore, the design of a purely automated pipeline gives the warranty of the reproducibility of the results.

The previously mentioned XCMS software for processing of MS is an example of a package distributed by Bioconductor, illustrating how different systems approaches have preferred the use of the R programming language.

Introduction: Part II, Malaria

Malaria overview

Malaria in humans is caused by five different parasites, *Plasmodium falciparum*, *P. vivax*, *P. ovale*, *P. malariae* and *P. knowlesi*. The scientific knowledge on the *P. vivax* biology if far less than what it is known for *P. falciparum*, however, the global population at risk is comparable (2.5 billion people live at risk of infection of *P. vivax* (Gething et al., 2012)) and it has been recently called as serious and threatening as *P. falciparum* infection (Baird, 2013; Quispe et al., 2014).

Life cycle of Plasmodium vivax

The malaria infection starts with the bite of an infected female mosquito from a susceptible species from the genus *Anopheles* that will release few parasites at the sporozoite stage. In the dermal tissue they are motile and move to small blood vessels that will allow them to reach the liver where they are engulfed by Küpffer cells to consequently penetrate a hepatocyte, the place where the parasite will differentiate to small trophozoite and after five days to a multinucleated schizont. By day six or seven the mature schizont will undergo a rupture to generate thousands of single nucleated merozoites that will find a way to get to the blood circulation where they invade reticulocytes (that are the young erythrocytes characterized by the expression of the $CD71^{+}$ marker (Malleret et al., 2014)) which comprise only 1-2% of circulating erythrocytes (Kitchen, 1938). Once in the infected red blood cell (iRBC, or the host cell), the merozoite starts the differentiation to an erythrocytic trophozoite, and remodels the host cell by creating a parasitophorous vacuole, cleft structures and caveolae-vesicle complexes (Aikawa, Miller, & Rabbege, 1975; Malleret et al., 2014). During the next 48 hours it feeds upon haemoglobin and divides around four times to

generate 12-16 new merozoites, the intermediate stage before the cellular division and after the nuclear division is also called schizont, only some of those schizonts will develop into gametocytes which are sexual stages that can survive in the invertebrate host closing the life-cycle.

Unlike *P. falciparum*, the *P. vivax* infected erythrocyte becomes quite deformable (Handayani et al., 2009; Suwanarusk et al., 2004) and promotes more rosetting events which is the adherence of uninfected erythrocytes to an infected one. The rosetting was historically thought as a mechanism to attract future hosts for the releasing merozoites reducing, this way, their time in the peripheral blood. However, it has been recently shown that in the case of the *P. vivax* infection, these rosetting complexes are mostly made up mature erythrocytes (also named normocytes)(E. Lee et al., 2014).

An additional characteristic of this malaria infection is the development of an additional life stage in the liver named hypnozoite, a dormant form that enables recurrent blood stages infections after months or even years from the first infection (Reviewed in (Mueller et al., 2009) and (Galinski, Meyer, & Barnwell, 2013). Further details of the biology of this parasite are beyond the scope of this thesis.

Malaria vaccination and treatment

Currently there are no licensed vaccines against malaria, despite more than 20 vaccine candidates in clinical trials; only one is in Phase 3 (RTS,S/AS01) which targets *P. falciparum* (Agnandji et al., 2012), leaving *vivax* malaria without any vaccine for the near future.

For the treatment of *P. vivax* infection, the recommended drug by the World Health Organization is the chloroquine in areas where it is still effective. In the countries where resistance has been confirmed the alternative treatment is the artemisinin-based

combination therapy, to eliminate the liver stage primaquine is used (Report, 2013)(World Malaria Report, 2013).

An additional problem associated with malaria eradication programs is the absence of, or a correct, molecular typing to measure incidence of the different *Plasmodium* species between asymptomatic people (Fru-Cho et al., 2014). This is a similar situation with the incorrect diagnosis of mixed infections as different species have different levels of parasitemia, explained by the different used cell hosts (Kitchen, 1938) (Mueller et al., 2009). Furthermore, interventions exclusively targeting *P. falciparum* in mixed infections are insufficient to eliminate the disease (Genton et al., 2008) and new species can be adapting to human cells as it was the case with the recently described naturally acquired human infection with *P. cynomolgi* (Ta et al., 2014).

Overview of the phospholipid metabolism in *Plasmodium* spp.

After the merozoite attaches, reorients and forms a junction, the parasite releases the content of rhoptry and microneme organelles which include proteins and lipids that participate in the formation of a parasitophorous vacuole membrane (PVM). The surface area of this structure corresponds to approximately 3% of the total area of the erythrocyte right after the endocytosis (Holz, 1977).

This initial contribution of biomass corresponds to a minimal proportion of the material needed to construct, maintain and increase in size (up to five times the content of phospholipids) the components of the PVM (Mitamura & Palacpac, 2003).

Given the importance of the phospholipid metabolism in the survival of the parasite, it represents a target for drug discovery research as well as an open field for the development of biomarkers *Plasmodium* species-specific.

The enzymes participating in the biosynthesis of fatty acids (FAs) can be divided in two categories, large multifunctional proteins (type I fatty acid synthases) are present

in mammals, fungi and some mycobacteria while the type II fatty acid synthases, where separate enzymes catalyse step by step the fatty acid production, are present in plants and most bacteria.

Species from the phylum Apicomplexa are unicellular Eukaryotes that parasitize vertebrates or invertebrates, some examples are the member of the genera *Babesia, Crystosporidium, Eimeria, Theileria, Toxoplasma* and *Plasmodium*. The evolutionary history of the Apicomplexa has its origins in two subsequent endosymbiosis events, a cyanobacterium was engulfed by a heterotrophic eukaryote that was subsequently taken up by an auxotrophic protist. The initial photosynthetic function of the organelle that initially was a cyanobacteria was lost as this organisms acquired a parasitic life-style keeping them away from sunlight (Reviewed in (van Dooren & Striepen, 2013)). The now organelle received the name of Apicoplast. With the uptake of additional genetic material encoded by that ancient cyanobaterium, some metabolic pathways were integrated to the host organisms in processes like energy transport and storage. However, some of the encoding genes were translocated to the nucleus where they are transcribed and then with the help of signal peptides they are targeted back to the apicoplast. Examples of these genes are acpP, fabH and fabZ which are involved in the type II fatty acid biosynthesis (Waller et al., 1998).

Mature erythrocytes are depleted of machinery used to synthesize new lipids and cannot endocytose, indicating that any changes observed in the membrane are products of degradation or are the result of metabolism of the infecting parasite (Lingelbach & Joiner, 1998). It is known that Plasmodium performs the *de novo* synthesis of phospholipids like phosphatidylcholine (PtdCho) from choline or phosphatidylethanolamine (PtdEtn) from Serine. Additionally, it has been shown that in *P. falciparum* the *de novo* synthesis of fatty acids takes place during the erythrocytic stage (Surolia & Surolia, 2001).

Aims of project

In contrast to *P. falciparum*, nothing is known about the lipid metabolism in *P. vivax* infected red cells. Our limited understanding of the lipidomic profile and general biology of *P. vivax* largely stems from the inability to continuously culture this species (continuous culture of *P. falciparum* was achieved in the 1970s (Siddiqui, Schnell, & Geiman, 1970)). Another hurdle worth mentioning has been the lack of knowledge on the metabolic differences between reticulocytes (of varying stages) and normocytes, in particular in terms of phospholipid composition (Malleret et al., 2013). This is important as *P. vivax* only invades early stage reticulocytes and never normocytes.

Given this background, this study seeks, first, to develop a bioinformatics pipeline, organized as an R-package, for the data analysis of MS and MS/MS experiments that can be applied to different sample comparisons. Secondly, to characterise the lipidomic profile of immature reticulocytes isolated from human cord blood cells (to provide us with the background lipid profile) to that of *P.* vivax-infected cord blood reticulocytes. Finally, to compare the observed changes in the *P. vivax* model with a *P. falciparum* model.

Materials and Methods

In this section, the first part summarizes the samples and previously done experiments that generated the data used for the analysis. The second part describes the procedures and functions written in R code to identify the candidates for the MS/MS experiment as well as the downstream processing of the information obtained by the MS and MS/MS experiments.

Description of the data.

Four *P. vivax*, and one *P. falciparum* clinical isolates were previously obtained from Shoklo Malaria Research Unit in northwestern Thailand by the group of Dr. Francois Nosten. The isolates were maintained in human cord blood cells following recently published protocols (Malleret et al., 2014) at the *P. vivax* laboratory of the National University of Singapore and processed for lipid extraction in two batches at the Singapore Lipidomics Incubator, at the National University of Singapore using published protocols (Narayanaswamy et al., 2014). For the *Experiment 1*, eight samples, described in Table 3, where identically processed.

Table 3

	Host cell	**Infecting parasite**	**Fraction on MACS**	**Cell count**	**# tech. replicates**
Pf1	RBC	*P. falciparum*	Negative	~90 mill.	2
Pf2	RBC	*P. falciparum*	Positive	~90 mill.	2
PvFE1	Cord Blood Reticulocytes	*P. vivax*	Negative	~10 mill.	2
PvFE2	Cord Blood Reticulocytes	*P. vivax*	Positive	~2 mill.	2
CBNorm1	Cord Blood Normocytes	NA	NA	~1 mill.	2
ReticFE1	Cord Blood Reticulocytes	NA	NA	~1 mill.	2
uRBC1	RBC	NA	NA	~90 mill.	2
BlankFE	NA	NA	NA	NA	2

Table 3. Description of the samples used for the first experiment. # tech. replicates, number of technical replicates; RBC, red blood cells; MACS, magnetic-activated cell sorting; NA, not applicable; mill, millions.

Pf1 and Pf2 are the negative and positive fractions from a cell sorter that separates DNA-containing cells (positive fraction) or not (negative fraction), respectively. PvFE1 and PvFE2 were separated using the same method. CBNorm1, ReticFE1 and uRBC1 are cell cultures that never had contact with parasites. Cord Blood Reticulocytes were sorted from normocytes by using the CD71 marker. Blank1 is a cell-free sample that followed the same lipid extraction protocol and will be used as a control for contamination during the lipid extraction process and the mass spectrometer performance. Cell count was used for normalization for the phospholipid extraction that was used as input in an Agilent 6550 iFunnel LC-ESI-QTOF-MS.

A second experiment (*Experiment 2*) enlarged the sample size exclusively for *P. vivax* infected cells. Table 4 describes the samples.

Table 4

	Host cell	**Infecting parasite**	**Parasitemia (%)**	**Total protein content**	**# tech. replicates**
Blank	NA	NA	NA	NA	1
Pv1	Cord Blood Reticulocytes	*P. vivax*	10.9	0.121	2
Pv3	Cord Blood Reticulocytes	*P. vivax*	15.1	0.136	2
Pv5	Cord Blood Reticulocytes	*P. vivax*	14.7	0.138	2
Reticulocytes	Cord Blood Reticulocytes	NA	0	0.120	3

Table 4. Description of the samples used for the second experiment. Parasitemia indicates the percentage or reticulocytes being infected. NA, not applicable.

The reticulocytes sample was exactly the same culture that was used to infect with three different clinical isolates of *P. vivax*, Pv1, Pv2 and Pv3.

Lipid extraction and Mass Spectrometry experiments

The protocol followed for the lipid extractions in both experiments were based on a non-targeted profiling of lipids, thus no internal standards were used. The first experiment focused on phospholipid recovery while the second one targeted phospholipids as well as sterols. All the samples were processed twice in the Mass Spectrometer, and in the case of the non-infected reticulocytes for the second experiment three times, to give technical replicates.

Data analysis

mzdata files from *Experiments 1* and *2* where processed in the same way except for a normalization step in the case of the samples where the cells were not counted (*Experiment 2*). There, normalization of the intensity values was performed based on protein estimation (See Discussion section for details about the normalization procedure chosen).

Feature detection and alignment of features

The standalone version of the software XCMS v1.38 (Smith et al., 2006) was used for feature detection with the centWave algorithm appropriate for high resolution LC/MS data sets (Tautenhahn et al., 2008), considering 30 parts per million as the maximal tolerated *m/z* deviation in consecutive scans, a range of 15 to 30 for the tolerated peak width, the default, obiwarp, method for retention time correction, a value of 0.015 for the width of the *m/z* windows used to group peaks and to create the peak density chromatograms and a deviation of 5 seconds as a maximum for the retention time.

Design of code to select candidates for MS/MS

Three types of data were present in both experiments, a negative control or blank, an infected sample and a non-infected sample. The general idea was to identify highly expressed features in a negative control, subtract them from the total set of features and then to identify the features being present with a significant higher intensity in the infected sample compared to the non-infected cells.

In detail, after the feature detection and alignment done by XCMS v1.38, the error rate is calculated per set of technical replicates. Assuming that a true and characteristic metabolite should appear in all the samples but considering some errors in the detection of the mass spectrometer and in the feature detection method, a threshold of 66% was used to eliminate those features lacking reproducibility based on technical replicates, (this filtering step was done by using *filterIntensity.R*, see Appendix B for full code).

The next step was the subtraction of false positives. To perform this, the fold change and p-value (two sided Student's t-test) are calculated using not normalized intensity values. Those features appearing 1.5 times or more in the blanks than in the rest of the samples as well as having a p-value equal or lower than 0.05 are considered as false positives and are discarded (see function *removeFalsePositives.R* in Appendix B).

For the case of the *Experiment 2*, as the cell count was not known, six different normalization strategies were tested in order to be able to compare infected and non-infected samples. 'Quantile normalization' normalizes the distribution of intensities per sample making each sample to have the same quantiles (Bolstad, Irizarry, Astrand, & Speed, 2003), this method is taken from the R package *limma* v3.18.13. 'Total Sum' assumes that the same lipid content is present on each sample so the total sum of intensities per sample should be the same. 'Median Fold by Sample' centers each distribution in the median of the fold changes, where the median values are obtained by sample, for technical reasons all zero values reported by the mass spectrometer are changed to 1. 'Median Fold by Feature' also centers the distribution in the median of

the fold change but obtaining the median values per feature, this follows the idea and code reported in (Veselkov et al., 2011), the authors suggested to use a replacement of 0.0001 in the zero values. 'Median Fold (limma)' this method also centers the distribution in the median of the fold changes so that each sample has the same median value, it follows the method called *normalizeMedianValues* from the *limma* package (v3.18.13) (Smyth, 2005). 'Protein Estimation' considered the values of protein estimation shown in Table 2 to make proportional the ratios in total sum of intensity values with the protein estimation per sample. These strategies are implemented in the function *featureNormalization.R* which can receive the columns of the blanks not to consider this data (see Appendix B for full code).

The same way the fold change and p-values are calculated for the blanks, these values are obtained for the infected cells versus the control samples (the filter is implemented in the function *fcpvalue.R*, see Appendix B for full code).

Two consecutive filters delimit the features in an *m/z* and retention time range that fits the lipid extraction protocol. For the *Experiment 1*, as mentioned earlier, it is expected to recover only phospholipids, then a lower *m/z* value of 400 was set, while for the *Experiment 2* the lower limit was set to 300. In both cases no upper limit for *m/z* was fixed. The range for retention time was set from 3 to 14 minutes. These parameters can be easily customized in the functions *mzmedFilter.R* and *rtFilter.R* as shown in Appendix B.

The next filter retains only those features being highly present in samples or in controls, named 'UP' or 'DOWN' features, respectively. The fold change cut off was set as 1.5 and the p-value as 0.05. The function implementing this is *UPorDOWNregulated.R* which generates two lists of features for each experiment for each mode (full code in Appendix B).

In high-resolution instruments like QTOF-MS, the 'exact mass' of a molecule can be calculated and peaks from different isotopes can be observed. With the attempt to obtain only one peak per molecule, the most abundant isotope will be sought. In the case of lipids, and as mentioned in the introduction section, the highest peak will coincide with the molecular mass resulting from the sum of the most abundant isotopes of each of its atomic constituents and this will be referred as the monoisotopic mass. To further simplify the analysis, only single-charged molecules will be considered. Then, the monoisotopic mass will correspond to the monoisotopic *m/z* value. Theoretically, at least the first isotope should be also present in the data in a smaller proportion as the monoisotopic peak. All these observations are taken into account in the *isotopeFilter.R* function which calls the *Isotope.R* function that retrieves the closest possible isotope to a given *m/z* and retention time values, considering a threshold, or error window, for *m/z* (0.015) and retention time (3 seconds). This function can be adapted to identify isotopes with either higher or lower mass, in other words, to find the [*M*+1] isotope assuming the given mass corresponds to a monoisotope, or the [*M*] when the [*M*+1] is given. It is also possible to specify if the feature that is being searched must have a higher or a smaller intensity compared to the query feature. Finally, an option to search for double charged features can be set, that is, features that are separated by a 0.5 value in the *m/z* dimension, considering the mentioned threshold.

Then, *isotopeFilter.R* creates a table with the features that correspond to monoisotopes. To do so, it discards features that appeared at the [*M*+1] or [*M*+0.5] position from another feature with higher intensity. Same for those without at least the [*M*+1] isotope. Actually, the features identified as coming from a double-charged metabolite are reported in a specific file. Figure 4 shows examples of the eliminated isotopic features.

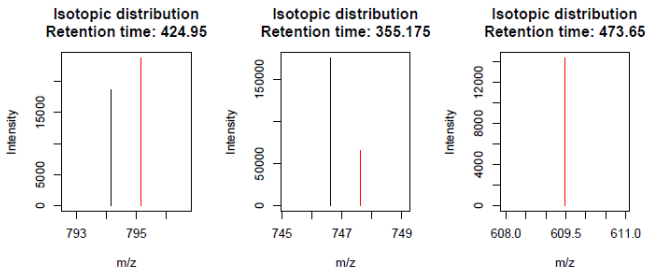


Figure 4. Three examples of features eliminated by the 'Isotope filter'. The bar in red is the one being eliminated while the black can be retained. From left to right: a possible monoisotope overlapping the [*M*+1] isotope as predicted by the intensity values; a possible [*M*+1] isotope, and a probable monoisotope lacking [*M*+1] or [*M*+2] isotopes.

The script used to call all the different functions and to produce the different graphs and tables containing intermediate outputs as well as the four final lists of candidates (two for each mode, one for each UP and one for each DOWN expressed features) is named *ObtainingCandidates.R* and the full code is present in Appendix B.

The only two dependencies of the described pipeline are the Bioconductor packages 'xcms' (Smith et al., 2006) and 'limma' (Smyth, 2005).

To sum up the designed pipeline for the identification of candidates for the MS/MS analysis Figure 5 shows the general flux diagram.

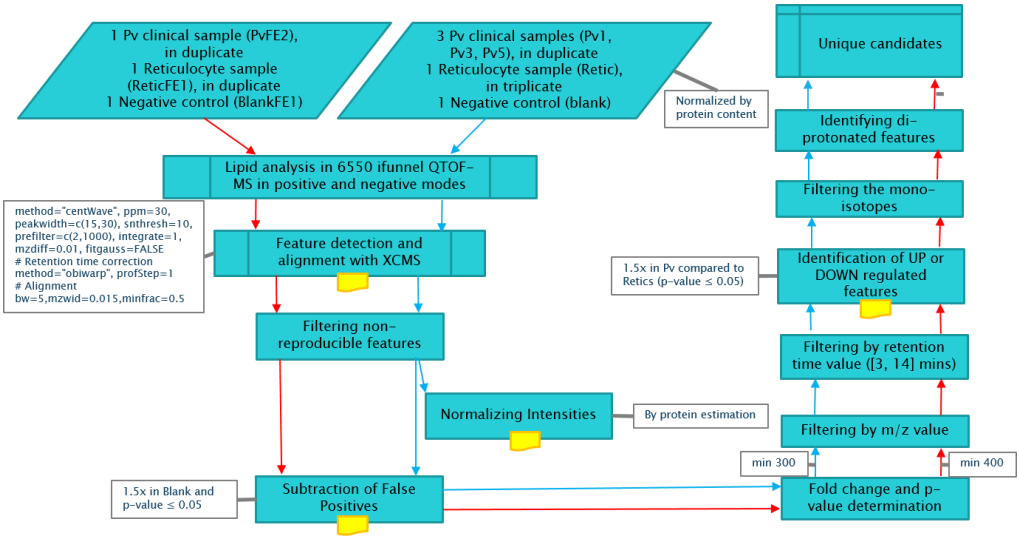


Figure 5. Flux diagram for the identification of differentially expressed monoisotopic features to proceed to MS/MS. Experiment 1: red arrows; Experiment 2: blue arrows. Yellow symbols indicate steps where intermediate tables are generated. White rectangles indicate parameters or normalization procedures.

Analysis of Tandem MS data: Lipid identification

mgf is the format of the files generated by a QTOF MS/MS instrument, one for each ionization mode and one for each input sample. This files contain sets of ion peaks defined by *m/z* and intensity values, one for each precursor (which in this case will correspond to one for each of the selected 'candidates') and can be present different times in the file with different retention times. When using the same column for the Liquid Chromatography as the one used for the MS experiment, it is possible to narrow down the number of spectra in order to select only those precursor ions analysed at the correct retention time, as this value will be constant with a margin to tolerate some technical variances. Thus, the first processing of the MS/MS data consists in selecting the spectra that corresponds to the *m/z* and retention time values of the candidates from the MS analysis in a window of 0.015 units for *m/z* and 30 seconds for the elution time. The function *extractingSpectra.R* will generate new mgf files with the spectra with the expected retention time (see full code in Appendix B).

The software NIST MSPepSearch (http://peptide.nist.gov/software/ms_pep_search_gui/MSPepSearch.html) is used for a high-throughput search of Tandem MS data against a given database of mass spectra. In this case, the LipidBlast database as well as custom libraries for Ceramides, Cholesteryl esters, Lysophosphatidylcholines, Phosphatidylcholines, Sphingomyelines and Lyso phosphatidic acids (Kind et al., 2013) were used, setting an *m/z* range of 200-1500, precursor ion tolerance of 0.015 and fragment peak *m/z* tolerance of 0.2, minimum match factor to output to 50.

The tsv files generated by MSPepSearch are then processed in R with the following functions: *getRTmins.R* parses the files to include a column with the retention time in seconds (See full code in Appendix B). The different tables using the same sample but different databases are joined with the function *concatenate.tsvs.R* (see full code in

Appendix B). Then, as the same precursor can have more than one spectra (because of measurements at different retention times), the hits are merged into a table with one line per precursor and the values of 'Lipid Class', Adduct, Total number of carbons in the fatty acyls, total number of double bonds in the fatty acyls, Lipid Subclass, and number of carbons in the first, second and third fatty acyl (if exists). As different values can be assigned to different spectra from the same precursor, this information is considered only when the reverse dot product value is equal or higher than 600, which has been the best performing algorithm for library search identification (Stein & Scott, 1994) (Kind et al., 2013). When there is more than one hit with a reverse dot product value higher than the threshold the information that is common between the high scored hits is kept. This procedure is implemented in the *LipidClassAssignment.R* function (for further details on this filter, such as the order in which the description of the lipid is assigned, see the full code and its corresponding comments in the Appendix B).

An optional step to identify 'well supported' lipids is to integrate information from both ionization modes, as some lipid species can acquire a positive or a negative charge like DG, PA, PG, PI, PS, PE, Cer, Inositolphosphoceramide and Mannosyl-inositolphosphoceramide (Ejsing et al., 2009). The function *clusteringSpectra.R* finds spectra from precursors with *m/z* values closer that a given threshold and determines if these spectra were assigned to the same lipid species (See Appendix B for further details and some comments about the clustering that uses the Adduct information).

Additional functions were written to facilitate the analysis and handling of mgf files. For example, in the tsv files generated by MSPepSearch there is a column named 'Peptide' with the name of the lipid species in a long word that contains the lipid class, the total number or Carbon atoms and double bonds, the adduct and ionization mode, the sub class and distribution of the carbons and double bond in the acyl chains (e. gr. 'PC 40:6; [M-Ac-H]-; GPCho(20:1(11E)/20:5(5Z,8Z,11Z,14Z,17Z))'). In order to separate this information the function *splitPeptide.R* receives the string and returns a

vector with the six mentioned arguments. This function is used in *LipidClassAssignment.R* to obtain these six values that will be compared between the different hits for the same precursor, the length of this vector can change to seven when the hit is a TG as it contains an additional fatty acyl.

Another function is *getMGF.R* which is a useful code to generate an mgf file with the spectra belonging to a given *m/z* and retention time values. Given that some handling of the data, like changing the class of the values in R or saving the tables in Excel can modify the length of the decimal digits by rounding them, a small variant of the function *getMGF.R* (*getMGF.feelinglucky.R*) can help in finding values for *m/z* and retention time that can be originated by rounding. These three new functions are in the Appendix B.

Finally, the filtered hits can be used to generate an mgf file for each of them with the function *generateMGFs.R* which receives the table with the filtered hits, the name of the sample and the mgf files produced by the Tandem MS experiment (See Appendix B for full code).

Analysis of Tandem MS data: Ion Lookup

For the alternative process of identifying lipid species by its similarity to a given database, a search can be done against a list of *m/z* values corresponding to well documented product ions that can give a hint on the identity of a lipid compound (Busik, Reid, & Lydic, 2009). In a similar fashion, data collected from Neutral Loss Scan can be used, but in this case the reported Neutral Loss is subtracted from the precursor ion and sought in the spectra of the MS/MS data. The list of values used for the scanning are summarized in Table 5, which was initially reported by (Busik et al., 2009) where they only presented nominal masses, here the list was improved by collecting the exact mass from the original reference, from databases or calculated. An

extended list is presented in the Table S1 in the Appendix A, with additional *m/z* values for Product Ions and Neutral Loss as found in the literature.

Table 5

Lipid Class	Mode	Precursor Ion	MS/MS Scan Type	Molecule	Exact Mass	Reference
SM, GPCho	Neg	[M+Cl]-	NLS	CH3Cl	50.488	Busik2009
SM, GPCho	Neg	[M+CH3OCO2]-	NLS	CH3OCO2H+(CH3)3N	135.1617	Busik2009
SM, GPCho	Neg	[M+CH3OCO2]-	NLS	CH3OCO2+(CH3)3NCHCH2	161.1989	Busik2009
SM, GPCho	Pos	[M+H]+	PIS	Phosphocholine	184.150662	Han2005
SM, GPCho	Pos	[M+Na]+	PIS	Sodium cyclophosphane	147.022	Han1995
GPCho	Pos	[M+Na]+	NLS	Sodium cholinephosphate	205.1404313	Han2005
SM, GPCho	Pos	[M+Na]+	NLS	Neutral phosphocholine	183.150662	Brugger1997
GPEtn	Pos	[M+H]+, M+Na]+	NLS	Phosphoethanolamine	141.063	Brugger1997
GPEtn	Neg	[M-H]-	PIS	Glycerol phosphoethanolamine derivative	196.0275	Han2005
GPIns	Neg	[M-H]-	PIS	Dehydrated phosphoinositol	241.104602	Han2005
GPIns	Pos	[M+NH4]+	NLS	Phosphoinositol+NH3	277.166282	Busik2009
GPSer	Pos	[M+H]+, [M+Na]+	NLS	Phosphoserine	185.072	Busik2009
GPSer	Pos	[M+Na]+	PIS	Sodium phosphoserine	208.0617693	Busik2009
GPSer	Neg	[M-H]-	NLS	Serine-H2O	87.0777	Han2005
CE	Pos	[M+NH4]+	PIS	Cholestane cation	369.3	Han2005
MG, DG	Pos	[M+NH4]+	NLS	H2O+NH3	35.0458	Busik2009
GPGro	Pos	[M+NH4]+	NLS	Phosphoglycerol+NH3	189.104222	Busik2009
GPA	Pos	[M+NH4]+	NLS	NH3+H3PO4	115.025702	Busik2009
GPGro, GPA	Neg	[M-H]-	PIS	Cyclic glycerophosphate derivative	153.058442	Han2005
GPIns	Neg	[M-H]-	NLS	Inositol unit -H2O	162.14058	Brugger1997
GPIns	Neg	[M-H]-	PIS	[H2PO4]-	96.9872	Brugger1997
SM	Neg	[M-H]-	PIS	Dimethyl-ethanolaminephosphate	168.108202	Brugger1997

Table 5. Exact mass values of Product Ion and Neutral Loss Scan experiments. SM, Sphingomyelin; GPCho, Glycerophosphatidylcholine; GPEtn, Glycerophosphatidylethanolamine; GPIns, Glycerophosphatidylinositol; GPSer, Glycerophosphatidylserine; CE, Cholesteryl ester; MG, Monoacylglycerol; DG, Diacylglycerol; GPGro, Glycerophosphatidylglycerol; GPA, Glycerophosphatidic acid; Pos, Positive ionization mode; Neg, Negative ionization mode; NLS, Neutral Loss Scan; PIS, Parent Ion Scan; Busik2009, (Busik et al., 2009); Han2005, (Han & Gross, 2005); Han1995, (Han & Gross, 1995); Brugger1997, (Brugger, Erben, Sandhoff, Wieland, & Lehmann, 1997).

Given that the presence of certain Product Ions of Neutral Loss can only suggest that the identity of a precursor is any of a subset of lipid classes, the identification of different ions is essential, and the more ions sustaining the same classification the more

likely the molecule was correctly assigned. As an example, the spectra from a precursor could contain an ion signal at an *m/*z=184.150662 which would suggest that the precursor is either a Sphingomyelin (SM) or a Glycerophosphatidylcholine (GPCho), and this is because both lipid classes can produce a Phosphocholine group that can be lost in an ion form. Thus, it would be necessary the identification of a Product Ion or a Neutral Loss that is exclusive for a SM and not for a GPCho, as it is the case of the Dimethyl-ethanolaminephosphate. The function *neutralLossScanner.R* will search for the listed Product Ions as well as ions resulting from the subtraction of the precursor *m/z* and the Neutral Loss values in the spectra generated in the MS/MS experiments. The hits will be stored and compared to identify either contradictions of confirmations for the assigned lipid class.

Finally, a list of lipid classes and associated frequencies will be reported for each precursor, this is expected as a single precursor can be measured several times in the Tandem MS experiment due to different elution times.

As it is possible that two lipids corresponding to different lipid classes have the same *m/z* value and, although much less likely, the same retention time, this is interrogated. This way a list of precursors with an unrelated second assigned lipid class is reported.

Additional details related to the algorithm used for the identification of the Product Ions or the Neutral Loss indicated in the Table 5 are present in the Appendix B together with the full code.

Figure 6 shows the general flux diagram of the designed pipeline for the lipid identification by using a lipid library or Precursor Ion and Neutral Loss Scan data.

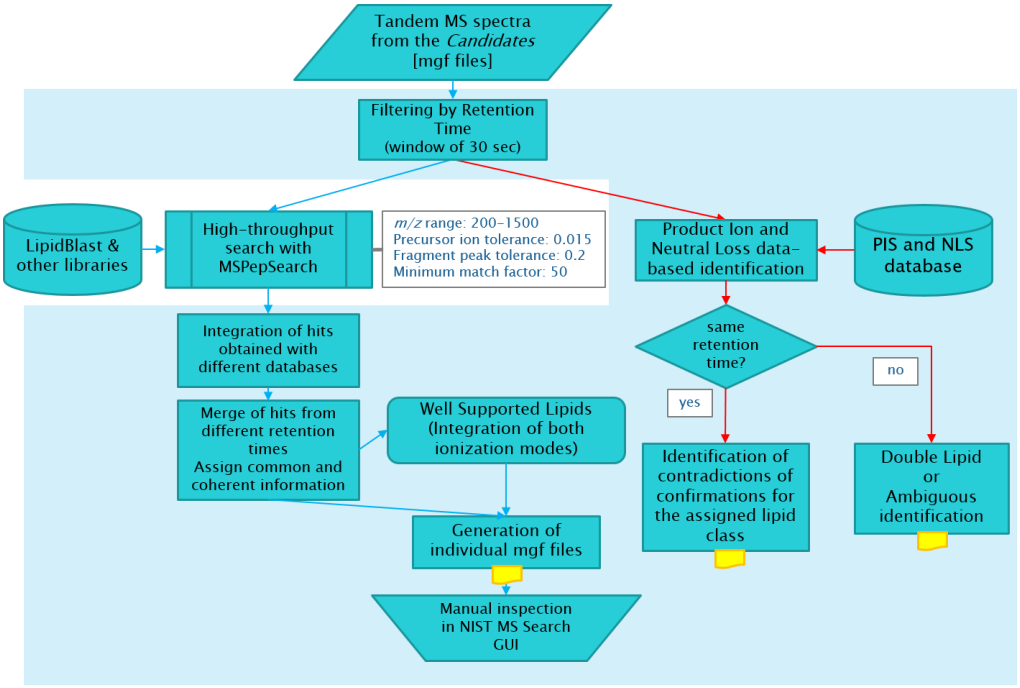


Figure 6. Flux diagram for the processing of Tandem MS data. Blue arrows: identification based on lipid libraries; red arrows: identification based on Product Ion (PIS) and Neutral Loss Scan (NLS) data. Yellow symbols indicate steps where reports are generated. White rectangle indicates parameters. Light blue background indicates data processing in R programming language.

Results

Mass Spectrometry Data Processing

Data obtained from four clinical samples of *P. vivax* were processed on an Agilent 6550 iFunnel QTOF-MS. Positive and negative ionization modes were analysed to obtain the *m/z* values corresponding to monoisotopic features being either statistically significant up or down expressed in *P. vivax* infected human cord blood reticulocytes when compared to non-infected reticulocytes coming from the same initial cell culture. The R-based pipeline used to identify those candidates is described in the Materials and Methods section and graphically depicted in Figure 5, the full code can be found in the Appendix B. The biological samples were processed in two different experiments. In the *Experiment 1* the cell count was obtained and used to determine the input volume for the mass spectrometer. For the *Experiment 2*, the protein content was used to determine the input volume for the mass spectrometer. A detailed description of the data is found in Table 3 and Table 4 as well as in the Material and Methods section.

The error rate was calculated per set of technical replicates, in all the cases the proportion of features that had all the replicates a value either above or below the threshold of 5,000 counts was higher than 93.6%, Figure 7, S1, S2 and S3 show the percentage of reproducible features for the *Experiment 1* and *Experiment 2* in positive and negative modes, respectively.

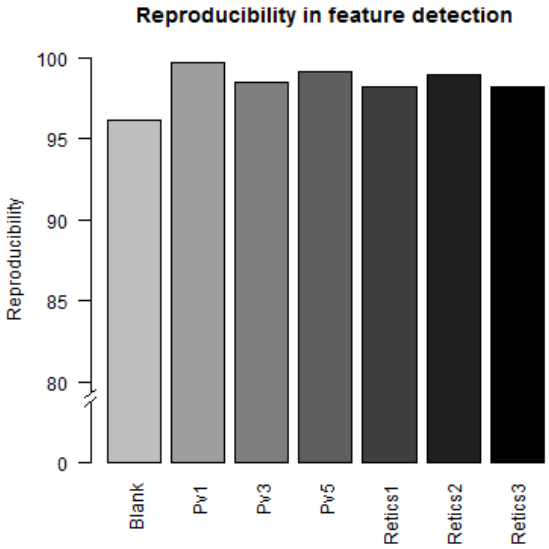


Figure 7. Reproducibility in feature detection in the *Experiment 2* in positive mode.

Evaluation of Normalization methods for Non-Targeted Mass Spectrometry Data

Due to the absence of cell count data for the *Experiment 2*, six normalization methods were tested in order to compare the samples:

- 'Quantile Normalization'. This is commonly used in microarray analysis and it is based on the idea that a straight line in a quantile-quantile plot of two distributions will indicate that they are the same, thus, it makes each sample to have the same quantiles (Bolstad et al., 2003).
- 'Total Sums'. Assuming that the same lipid content is present on each sample including those infected with the malaria parasite, the total sum of intensities per sample should be the same.
- 'Median fold normalization by sample'. It centers each distribution in the median of the fold changes where the median values come from the samples. This idea came out as a slightly modified version of the following widely used method but here assuming that the large number of lipids will make the global lipid expression per sample similar.

- ‘Median fold by feature’. As already mentioned, it centers the distribution of intensities in the median fold of the fold change calculated by feature, this follows the idea and code reported in a metabolic study using UPLC-MS (Veselkov et al., 2011).
- ‘Median fold (*limma*)’. It also makes each distribution to have the same median, this method was proposed by (Smyth, 2005) to work with microarray data and it is now the standard for this kind of data.
- ‘Protein estimation’. It assumes that the protein content in infected and non-infected cells remains the same, then the estimated protein content is used to normalize the sum of the intensities per sample.

In an attempt to evaluate which normalization procedure was the best for the present data, all these methods were tested. As mentioned in the Materials and Methods section, there is no consensus on the best method to assess the validity of the normalisation process, therefore some ideas were borrowed from tried and tested microarray or metabolomic protocols (Bolstad et al., 2003; Smyth, 2005; Veselkov et al., 2011). Following the idea that a normalization procedure should reduce the standard deviation (SD) of the measurements between samples coming from the same type (i.e. infected or not infected), the boxplots in Figure 8 intend to show the compression of the SD as a result of the different methods. While the ‘median fold normalization by sample’ practically removed the outliers, not a big difference was observed using this approach.

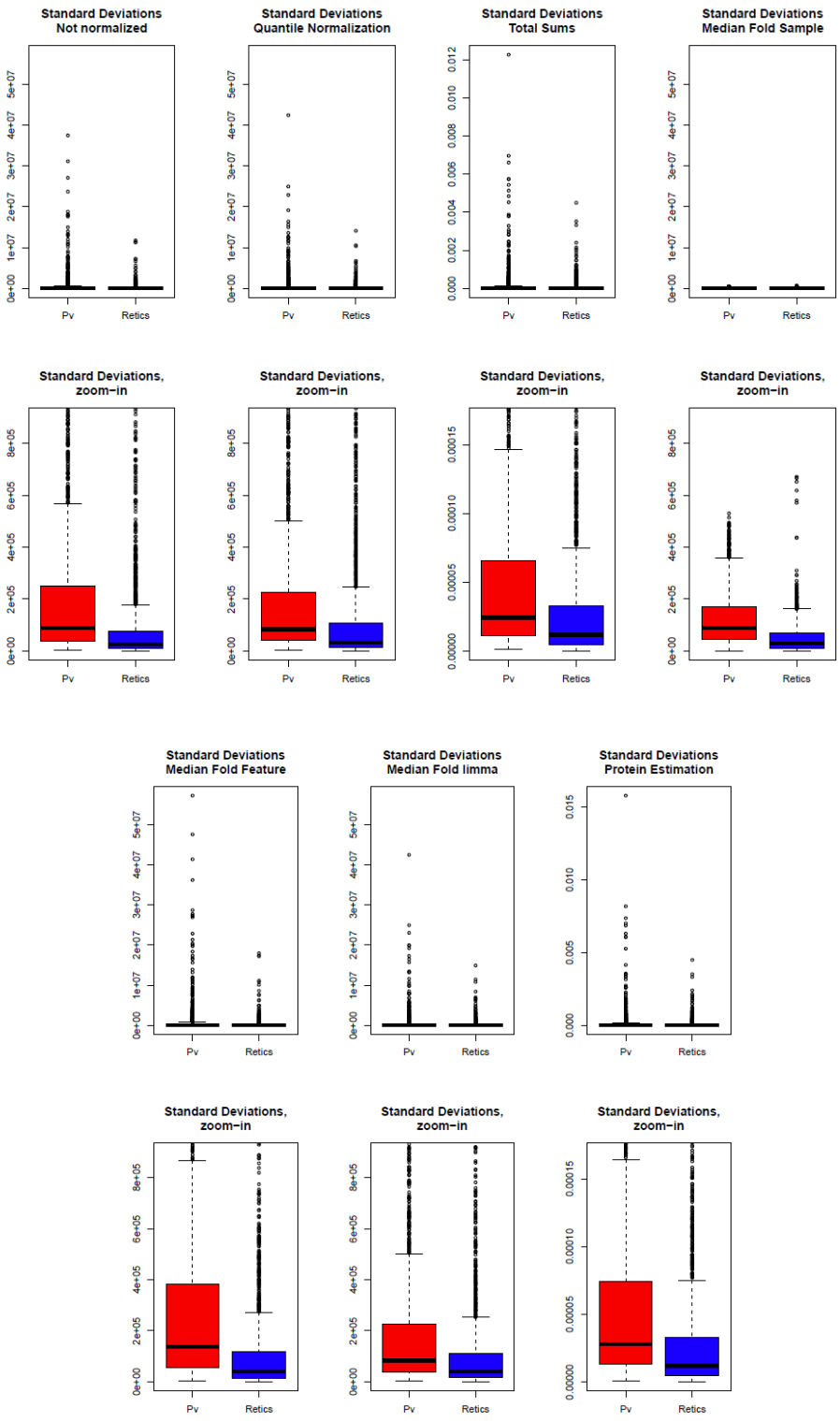


Figure 8. Boxplots of the standard deviations (SD) for each m/z value observed separated by conditions (infected (Pv) or not infected (Retics)). For the 'total sums' and 'protein estimation' methods a different scale was used as in these two cases the maximum is set to one.

Given that technical replicates should show variation due to noise during the measurements, an effective normalization method should group them together, Figure 9 shows that only normalization by protein estimation gets this clustering.

Finally, a more common way to evaluate the improvement of a normalization method is the usage of a multiparametric analysis (J. Lee et al., 2012; Veselkov et al., 2011). Here, the six methods as well as the original data were evaluated with PCA (using the function *prcomp()* from the *stats* package in zero centered mode), inspecting the clustering of the technical replicates by observing the plots of the first and second principal components they show that the normalization by 'protein estimation' clusters the non-infected samples all together while infected samples get grouped by clinical isolate (Figure 10 and Supplementary Figures 4-8).

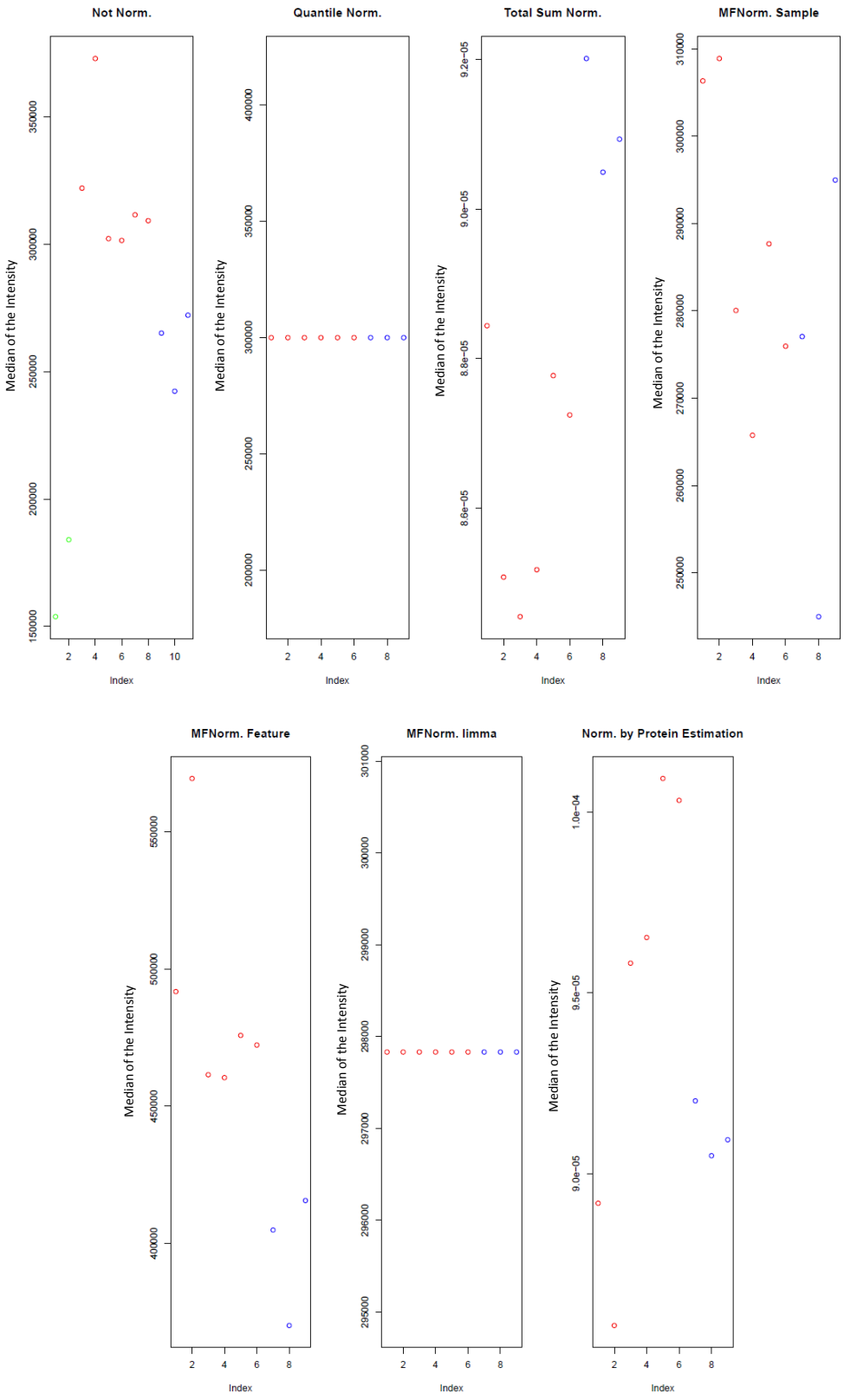


Figure 9. Plot of the median values of the intensity for each of the samples using the six normalization methods and the not normalized data ('Not Norm.'). Plotting the median helps choosing a feature behaving in a common fashion, avoiding considering a contamination metabolite or an artefact. Green, blank; red, infected samples; blue, non-infected cells. The order in the X-axis is the following: blank replicate 1 (r1), blank replicate 2 (r2), Pv1 r1, Pv1 r2, Pv3 r1, Pv3 r2, Pv5 r1, Pv5 r2, reticulocytes r1, reticulocytes r2, reticulocytes replicate 3. The normalization was not performed for the blank samples.

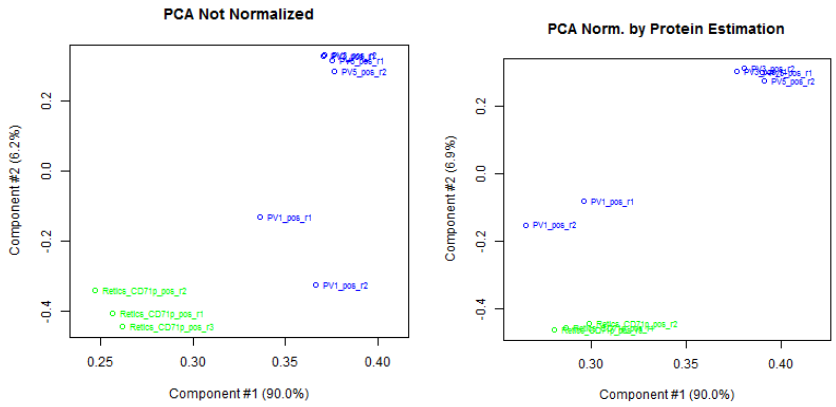


Figure 10. Principal component analysis original and normalized data by protein estimation using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.

Identification of Ions Differentially Present

By using the normalized data by 'protein estimation', the pipeline described in the Materials and Methods section was used with the data from the *Experiment 1* and *Experiment 2* in positive and negative modes. Figure 11 and Figure S9 depict the flux diagram of the filtering protocol that was followed to distinguish the statistically significant monoisotopic features in positive and negative modes, respectively, including the different numbers of filtered features at each step. A total of 160 features were predicted as monoisotopic candidates being more abundant in *P. vivax*-infected human reticulocytes than in non-infected cells from the same type (*p-value*≤0.05, Student's t-Test), while 76 were less abundant (*p-value*≤0.05), considering the positive and the negative mode together. Only one feature was identified as di-protonated. Table 6 shows the number of predicted monoisotopic features for each mode, condition and experiment. These features made up the candidates for the MS/MS experiment.

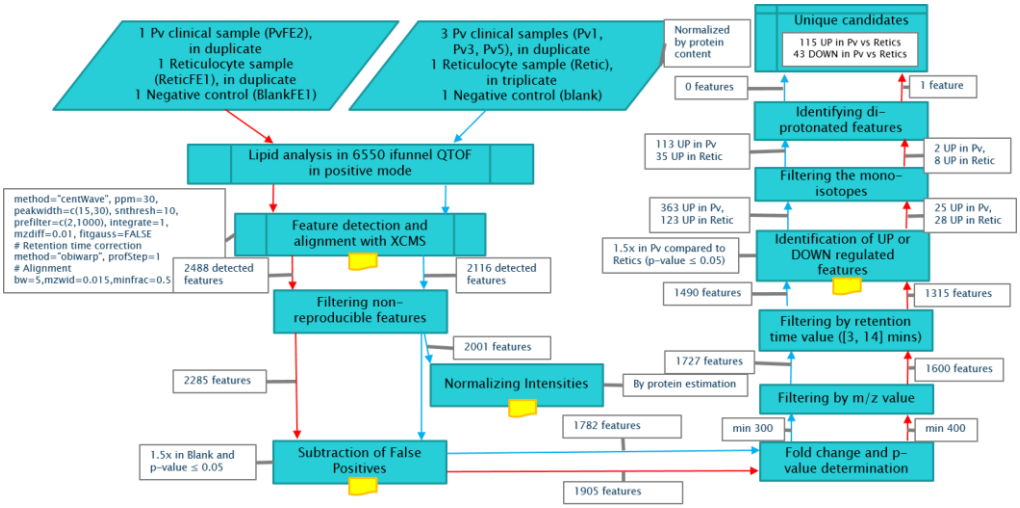


Figure 11. Flux diagram including the number of filtered features in each step in the positive mode.

Table 6

		Normalized by Protein Estimation		Not Normalized		
Mode		***P. vivax*, *Experiment 2***	**CB Reticulocytes, *Experiment 2***	***P. vivax*, *Experiment 1***	**CB Reticulocytes, Experiment 1**	
	compared with:	CB Reticulocytes, *Experiment 2*	*P. vivax*, Experiment 2	CB Reticulocytes, *Experiment 1*	*P. vivax*, *Experiment 1*	
Positive	p-value ≤0.05	113	35	2	8	**158**
Negative	p-value ≤0.05	40	30	5	3	**78**
		153	**65**	**7**	**11**	**236**

Table 6. Summary of the number of candidates being more abundant in infected (*P. vivax*) or not (CB Reticulocytes) in each mode and separated by experiment (*p-value*≤0.05, Student's t-Test).

To assess if choosing a different normalization method would give a different set of candidates, the whole pipeline was run for the six normalization methods as well as with the raw (original) data and the overlapping was calculated. Table 7 shows that a very small difference existed based on normalization procedures.

	Candidates UPPv	Candidates UPRetic	Total
Not Normalized	128	12	140
Quantile Normalization	100	58	158
Total Sum	98	64	162
Fold Change Sample (LB)	128	12	140
Fold Change Feature (hpbenton)	128	12	140
Fold Change limma	106	44	150
By Protein Estimation	110	35	145

99

Intersection

contained

Table 7. Comparison of the number of features predicted by using different normalization methods and the original data for the positive ionization mode. UPPv, up regulated or more abundant in the infected samples; UPRetic, more abundant in non-infected samples. The square brackets indicate pairs of compared sets that either share an intersection with the depicted number of elements on it or that are one contained in the other, for example, the intersection of the Quantile normalization UPPv and Fold change using limma package contains 99 elements and this last set is fully contained in the protein estimation set.

Further analysis of the candidates revealed that there is a difference of 18 features that are not considered as significant more abundant in the malaria samples when the pipeline is run without any normalization procedure in the positive ionization mode (see Table 7). In order to understand why some features are eliminated by the normalization by Protein Estimation, that otherwise would remain in the list of candidates if no normalization procedure was used, 10 random *m/z* values corresponding to the set of candidates obtained when using protein estimation and another 10 random *m/z* values from the set of 18 candidates being only predicted when no normalization method is used were selected to compare the median of their intensity values. Figure 12 shows in the upper row the 10 chosen *m/z* values that were only obtained when no normalization was applied (eliminated) and in the lower row the 10 chosen *m/z* values obtained after either normalization by protein estimation or no normalization. The two plots at the left show the intensity values before normalization and at the right after normalization by protein estimation. The colors of the open circles indicate if the sample is a blank (red), a *P. vivax*-infected reticulocyte (blue) or a non-infected reticulocyte (green). Lines connect measurements with the same *m/z*. In the plots it can be observed that at least one feature from the 'eliminated' set behaves more

irregularly even after the normalization while the behaviour of the 10 selected features in the 'common' candidates is more homogeneous and a clear difference is observed with respect to the non-infected samples. This indicates that the normalization by Protein Estimation is correctly eliminating features that do not have a homogenous pattern of differencial intensity between infected and non-infected samples.

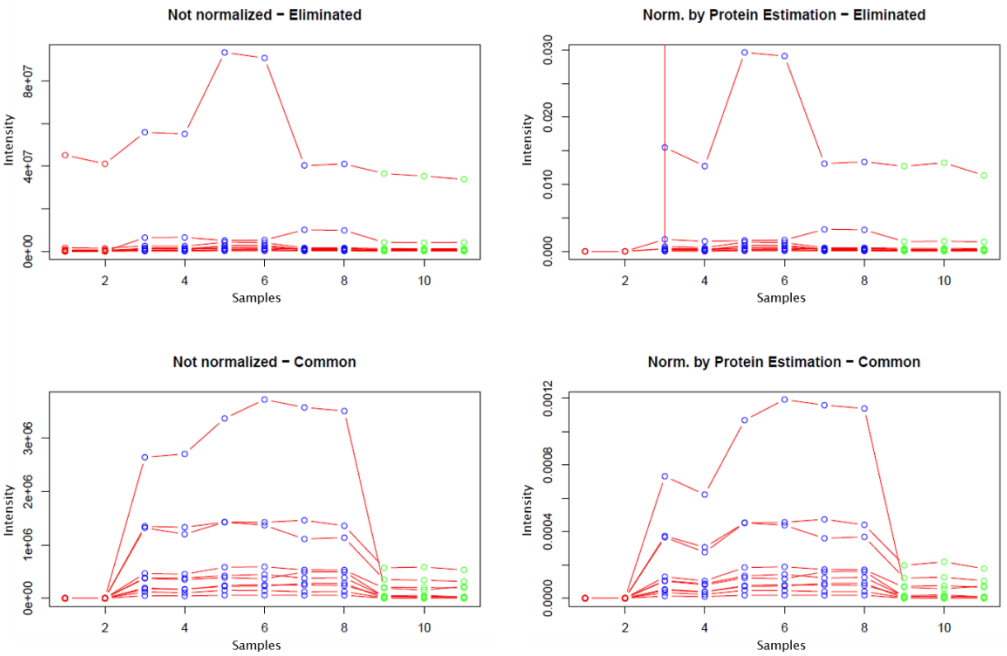


Figure 12. Comparison of the distribution of intensities for specific m/z values before and after normalization by protein estimation. Upper row, ten randomly selected *m/z* values followed in the different samples (red lines) corresponding to predicted candidates when the not normalized data was used but not predicted when protein normalization was used; lower row, ten randomly selected *m/z* values corresponding to predicted candidates in both not normalized and normalized by protein estimation; left column, the intensity values before normalization; right column, intensity values after normalization by 'protein estimation'. The colour of the circles indicates: red, blank; blue, infected samples; green, non-infected samples. Circles joined by the red line represent features with the same *m/z* value in different samples.

For the presented reasons, the normalization method based on 'Protein Estimation' was selected as the best procedure to reduce the variation due to factors that are irrelevant with regard to the infection process, such as the variations in the lipid extraction of the detection of the instrument, and it was applied to the samples from the *Experiment 2* from both infected and non-infected cell samples. Figure 13 and Figure S10 show a sum up of the 236 candidates identified in both E*xperiments* in positive and negative

ionization modes, respectively. 42 of those features (26% of the ions overrepresented in *P. vivax*-infected samples) were absent in the signals obtained in the non-infected samples. The list of the 236 candidates was used for a Tandem MS experiment.

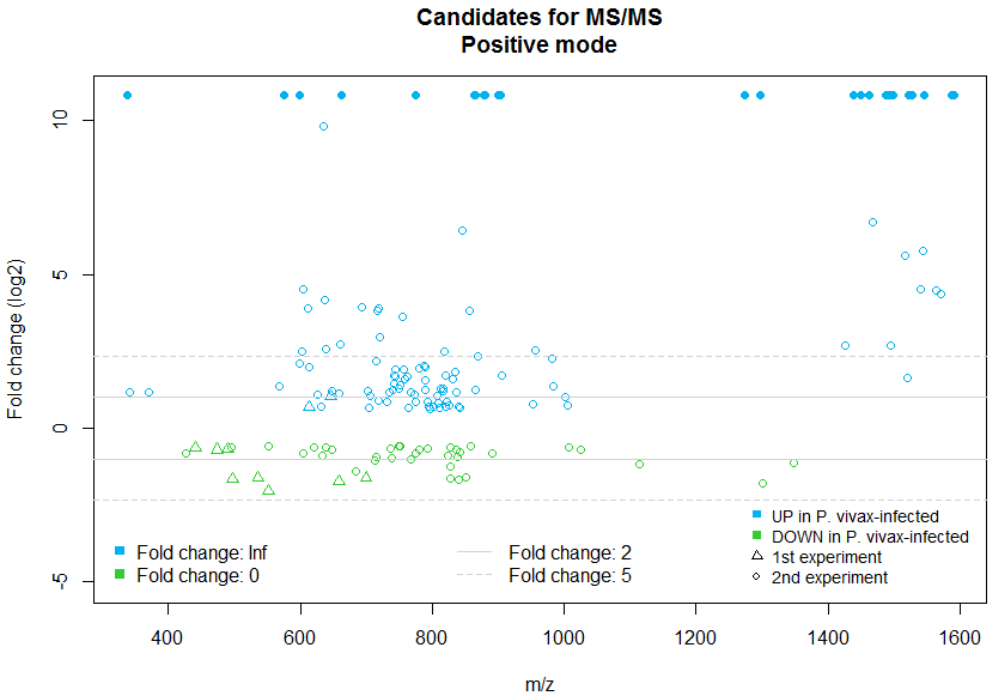


Figure 13. Distribution of the candidates for MS/MS in positive mode. The solid blue symbols denote an infinite fold change obtained when no signal existed in the non-infected sample.

Tandem Mass Spectrometry Data Processing

The list of candidates obtained from the MS analysis was used for Tandem MS on an Agilent 6550 iFunnel QTOF-MS in both positive and negative ionization modes using the same lipid extraction from the samples of the *Experiment 2* as in the MS experiments. The QTOF-MS generated *mgf* files that correspond to the input data for the following analysis.

More than 32 thousand spectra were detected but less than 5% of them had a similar retention time with the initial measurement obtained during the MS experiment. A

window of 30 seconds was used to compensate technical errors, Table 8 describes the raw and filtered data.

Table 8

Condition	UP in infected reticulocytes		UP in non-infected reticulocytes	
Ionization Mode	Positive	Negative	Positive	Negative
MS/MS output	11,271	11,173	4,859	4,843
Filtered by Retention Time	574	530	237	236
Hits with MS/MS libraries	57	281	9	26
Precursors with assigned LC	15	66	3	9
Mode/Adduct clustering	39		8	

Table 8. Summary of the number of spectra obtained from the Tandem MS experiment (MS/MS output) and after filtering the spectra with Retention Times that do not coincide with the time reported during the MS experiment (a margin of 30 seconds is used to tolerate technical errors). 'Hits with MS/MS libraries' refer to the number of hits retrieved with MSPepSearch in high-throughput mode. 'Precursors with assigned LC' describes the hits with non-ambiguous assigned Lipid Class. The number of different precursors regardless the mode or adduct measured is shown in the last row.

Lipid Identification by Library Search

The filtered mgf files were processed with the NIST software MSPepSearch for high-throughput search of spectra corresponding to a reported lipid within a library. The computer-generated LipidBlast database was used as target and the results were merged with other smaller custom libraries for specific lipid classes like Cer, CE, Lysophosphatidylcholines, Phosphatidylcholines, SMs and Lyso-PAs (Kind et al., 2013), as described in the Materials and Methods section.

The hits produced by this tool (373 counting all the filtered spectra and including possible multiple hits from the same query) are, afterwards, processed in order to eliminate low quality hits (based on the reverse dot product calculated by the MSPepSearch software) or queries whose assigned lipids were discordant at the lipid class level between the spectra with the same precursor and exactly same retention time (which means that the same query was assigned ambiguously). When the lipid class is concordant but the subclass or the distribution of carbons in the acyl chains is discordant, the shared (concordant) information is kept.

To take advantage of the information contained between the two ionization modes, lipids assigned to precursors in more than one mode are identified. To perform this, a clusterization method was developed. The subclass, total number of carbons, total number of double bonds and the distribution of carbons and double bonds in the acyl chains of the hits with the same *m/z* value (or with a difference equal or less than 0.02 units in the *m/z* value) are compared. At this point the information of the adduct will also be used, to give chance that one spectra had a hit with a different adduct, hence, having different hits the same retention time and *m/z*. In the case a second hit has the same mode and adduct, one of them is eliminated as no additional information will be provided (that was just a case of a delay in the elution of the lipid compound in the LC column). On the other hand, if a second hit is found coming from a different mode or adduct this is reported as it is a stronger confirmation of the identity of the lipid. In the case of the *P. vivax*-infected and non-infected data, no second hits were obtained, 8 and 39 different precursors from the non-infected and the *P. vivax*-infected set were assigned to a lipid class, respectively (47 spectra when including assigned lipids in different modes or adducts). An individual mgf file was generated for each one of these spectra.

Once the large number of initial measurements was narrow down to up to 0.15%, a manual curation of the hits was performed using the graphical user interface of MSPepSearch to verify the concordance of the peaks in the MS/MS spectra and the reported peak distribution in the different libraries. The exclusion criteria included the deletion of hits where one or both of the acyl chains did not have a match or if it could be ambiguously assigned to a second lipid species. An example of the match found between the obtained spectra and the library is shown in Figure 14, where the four most abundant peaks of the spectra at precursor *m/*z=816.5769 $[M\text{-Ac-H}]^-$ and retention time 8.80 min had perfect match with GPCho(16:0/18:2). The peaks corresponding to the

fragments with masses 742.5386, 279.2322, 255.2322 can be observed in Figure 14 and are also listed in the last column of Table 9.

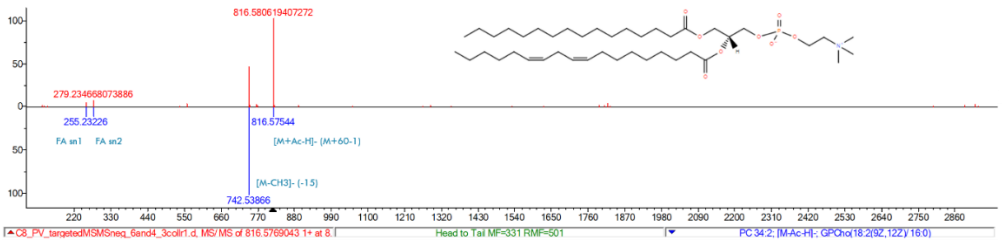


Figure 14. The spectra obtained from the MS/MS experiment at *m/z*=816.5769 and retention time 8.80 min (top) matched the spectra of GPCho(16:0/18:2) (bottom). The structure of the lipid with the double bonds in an arbitrary position is also depicted.

A total of four MS/MS spectra were unambiguously assigned to lipid species within the sample group corresponding to non-infected reticulocytes while nine were assigned within the sample corresponding to *P. vivax*-infected reticulocytes. The identified lipids are listed in the Table 9. Interestingly, two of the assigned precursors where not identified in the non-infected sample as observed in Figure 15.

Sample	Precursor (*m/z*)	Retention time (min)	Lipid	MS/MS
Non-infected reticulocytes	836.5442	8.34	GPSer(18:0/22:5)	749.5121, 483.2513, 465.2407
	810.5298	7.97	GPSer(20:4/18:0)	723.4964, 457.2356, 439.2251
	834.5294	7.93	GPSer(18:0/22:6)	747.4964, 481.2356, 463.2251
	838.5598	8.45	GPCho(20:5/16:0)	764.5230, 301.2166, 255.2322
***P. vivax*-infected reticulocytes**	816.5769	8.80	GPCho(18:2/16:0)	742.5386, 279.2322, 255.2322
	864.5752	8.53	GPCho(20:4/18:2)	790.5386, 864.5754, 303.2322
	830.5917	9.52	GPCho(18:0/18:2)	770.5699, 830.5911, 283.2635
	758.5849	8.77	GPCho(16:0/18:2)	184.0738, 758.5699, 699.4964
	892.6052	9.48	GPCho(18:0/22:6)	818.5699, 892.6067, 305.2479
	840.5775	8.71	GPCho(20:4/16:0)	766.5386, 840.5754, 279.2322
	794.6160	9.51	PC(P-18:0/20:4)	526.3298, 508.3768, 467.2563
	575.5221	7.68	DG(16:0/16:0/0:0)	263.2560, 575.5223, 337.2928
	844.6084	9.04	GPCho(18:0/18:2)	844.6068, 770.5699, 283.2635

Table 9. Identified lipids by MS/MS analysis. Four and nine lipid species were identified in the MS/MS data from the non-infected and *P. vivax*-infected reticulocytes, their *m/z* value and retention time are also shown. The last column indicates the first three most abundant fragments used to identify the lipid. GPCho, Glycerophosphatidylcholine; PC, Plasmenylphosphatidylcholine.

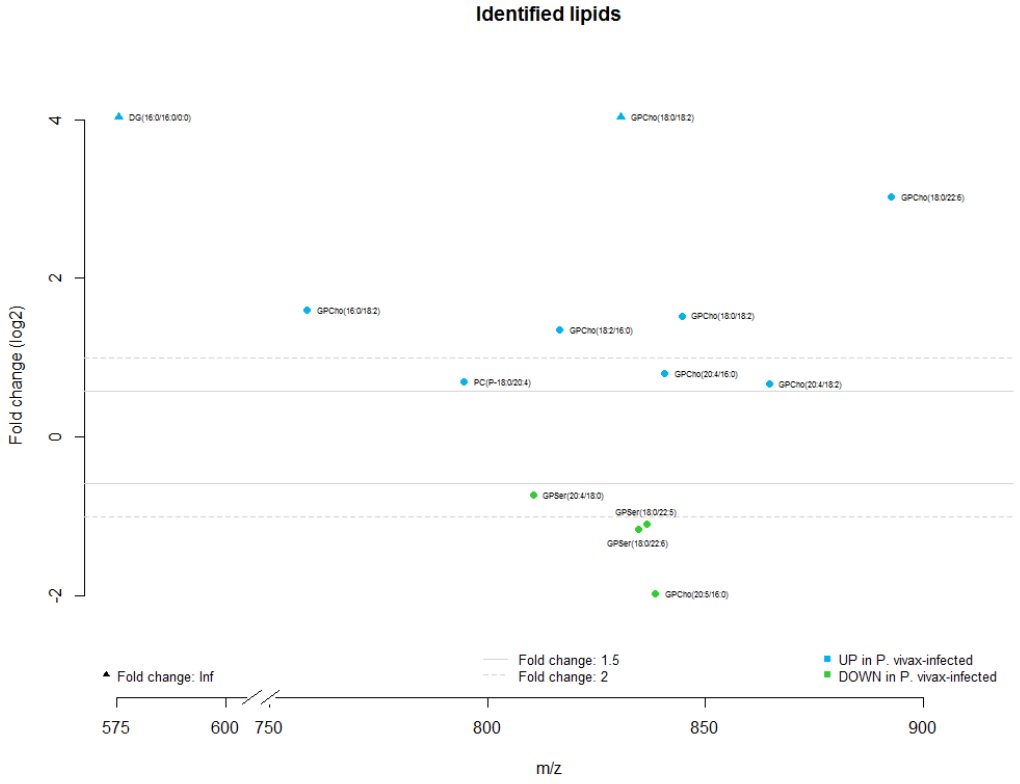


Figure 15. The identified lipids in from the MS/MS experiment. Nine and four lipids were identified in the *P. vivax*-infected and non-infected reticulocyte samples, respectively. Their *m/z* value and fold change as found in the Experiment 2 are illustrated. In the case of DG(16:0/16:0) and GPCho(18:0/18:2), that were not identified in the non-infected samples, they are drawn at an arbitrary position.

Lipid Identification by Product Ion and Neutral Loss Data

In order to expand the search for possibly new lipids within the analysed samples, a broader search was performed by looking at ions that are characteristic of certain lipid species. For example, the loss of phosphocholine is common after the fragmentation of SM or GPCho, the sodium cholinephosphate is only characteristic for GPCho fragmentation spectra and the phosphoetanolamine is generated only when a GPEtn is processed in MS/MS. Thus, a list of the masses of ions and neutral molecules that are commonly generated after fragmentation of lipids was collected as described in the Materials and Methods section. These values are used by a new algorithm to identify the possible lipid class for the different filtered spectra corresponding to the analysed samples. Table 10 summarizes the different precursors assigned to a single Lipid Class and some that remain ambiguous. It is worth noting that some classes where exclusively

found in one or the other sample. Tables S2 and S3 show the *m/z* values of the precursors and its corresponding assigned class as well as the 'Supporting value' which indicates how many times the same precursor was assigned to more than one Product Ion or Neutral Loss corresponding to a different Lipid Class because it was identified in different ionization modes or a different retention times.

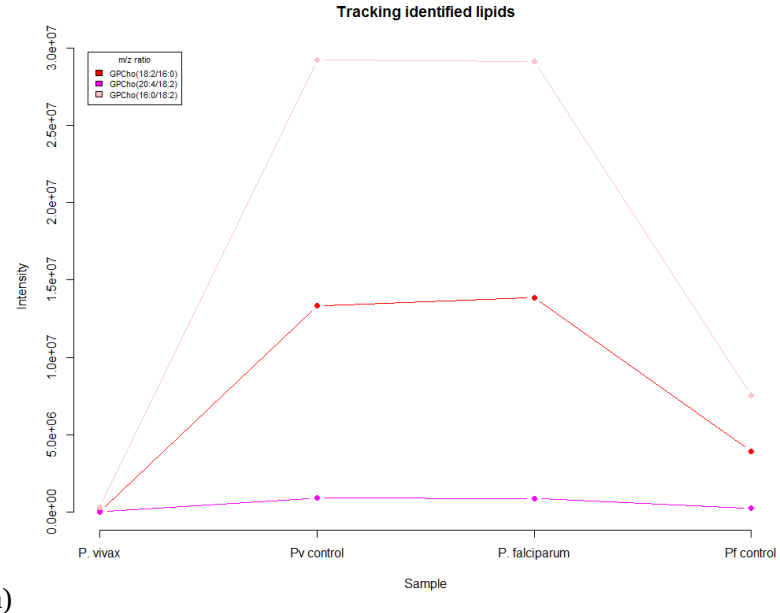
Possible Lipid Class	Non-infected Reticulocytes	*P. vivax*-infected reticulocytes
SM	2	0
GPIns	1	1
GPSer	6	2
GPEtn	1	8
GPA	0	2
CE	1	0
MG, DG	3	5
SM, GPCho	3	15
Gro, GPA	0	2
Total	17	41
Ambiguous	0	6

Table 10. Number of precursors identified at the Lipid Class level by Product Ion and Neutral Loss data-based assigning. When the same precursor was assigned to more than one Product Ion or Neutral Loss corresponding to a different Lipid Class it was considered as 'Ambiguous'. SM, Sphingomyelin; GPCho, Glycerophosphatidylcholine; GPEtn, Glycerophosphatidylethanolamine; GPIns, Glycerophosphatidylinositol; GPSer, Glycerophosphatidylserine; CE, Cholesteryl ester; MG, Monoacylglycerol; DG, Diacylglycerol; GPGro, Glycerophosphatidylglycerol; GPA, Glycerophosphatidic acid.

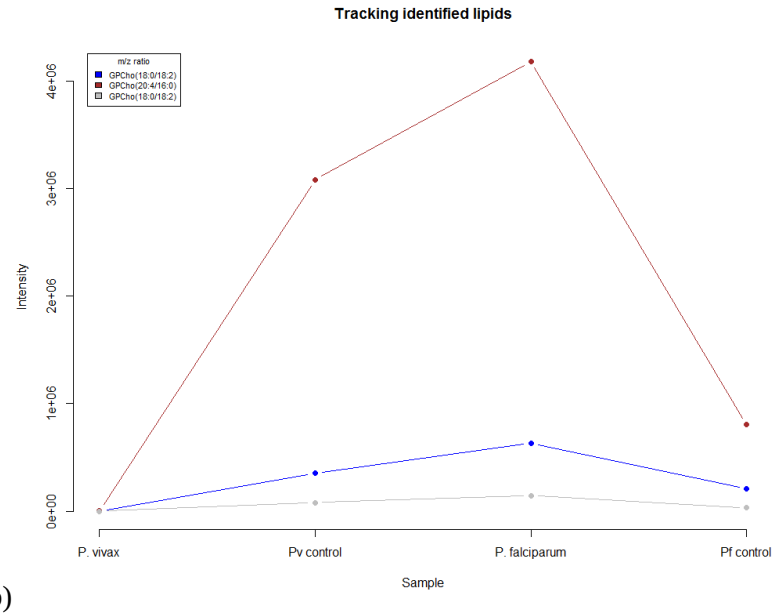
Tracking of Identified Lipids

The identified lipid species were sought within the data from the *Experiment 1* as a proof of principle of the useful combination of information collected from different experiments. Figure 16 shows that different behaviours can be observed in different maturation stages of erythrocytes and with different *Plasmodium*-infected samples. These results should be taken with caution as they correspond to a single experiment that requires further validation. Therfore, they can be inconsistent with the observed patterns from the lipid extractions from the clinical samples (*Experiment 2*) as observed in the two identified lipids over represented in the non-infected samples from the

Experiment 2 that did not show the same abundance ratio in the *Experiment 1* (see Supplementary Figure 11 for further details).



a)



b)

Figure 16. Tracking of the identified lipids being over-represented in the *P. vivax*-infected samples as found in the *Experiment 2* in the data from *Experiment 1*. Two different behaviours can be observed, similar abundance in P. falciparum-infected sample as in the non-infected reticulocyte (a) and higher abundance in *P. falciparum*-infected sample when compared to non-infected reticulocytes. In the X-axis, the samples are as follows: *P. vivax*-infected reticulocytes, non-infected reticulocytes, *P. falciparum*-infected erythrocytes, non-infected erythrocytes.

Discussion

Contributions

In the presented work three contributions can be highlighted, (i) a robust (the implementation can capture several errors in order to prevent the program to die) and flexible (it is possible to use different number of samples and controls as well as blank samples, all parameters can be easily adjusted in the running line and sub-routines can be prevented to be executed if desired) pipeline for the analysis of MS and Tandem MS data was developed for their easy use in R language with the option to receive the very raw data generated by the mass spectrometers to perform the whole analysis in a reproducible manner preventing many possible technical details that could face the analysis, such as different number of samples per condition, variations in retention times between the MS and the MS/MS experiments or the presence of multiple lipid classes having the same *m/z* and retention time. It is also able to be adjusted at different times of the analysis, for example, it can receive different number of samples per condition, including 'blank' samples that can measure the level of noise in the mass spectrometer. The width of the window for the *m/z*, retention time values, the percentage of samples expected to have an up or down expressed feature in order to be considered as a true differentially expressed ion and many other parameters can be modified depending on the kind of instrument used or the comparison intended to be done. This implementation is only dependent on two R packages: *xcms* and *limma* and the MSPepSearch tool, being the rest newly generated code.

(ii) In terms of the malaria field, the design of the pipeline allows the correct comparison of unique samples with their correct control. Given the particular preference of *P. vivax* parasites (and most likely total dependence) for reticulocytes (Kitchen, 1938), it stands out the importance of using non-infected reticulocytes as the

control for the infected cells, instead of the heterogeneous red cells. All this, without losing the track of the identified molecules in a control cell group of uninfected red blod cells (uRBC) and in other *Plasmodium* species (as in this case, the Pf1 and Pf2 samples).

Finally, (iii) the congruent findings with the reviewed literature represent the proof of concept of the efficacy of this pipeline and give an idea of the sensitivity that can be obtained by working with untargeted MS and Tandem MS experiments with lipid extractions from malaria infected samples and their controls.

The identified lipids

This is the first approach reported so far trying to characterize the differences in the lipid composition of the reticulocytes before and after the infection with *P. vivax*. Although quite reduced in the number of samples to reliably probe the metabolization or the synthesis of particular lipid species, this initial observations suggest the metabolization of some GPSer molecules to GPCho. Seven out of the nine identified lipids significantly over represented in the *P. vivax*-infected samples corresponded to GPCho lipids with 16 to 22 Carbons in the fatty acyls having from zero to six double bonds. The over-representation of GPCho in the identified metabolites is in agreement with the described enrichment of the same molecule in *P. falciparum*, that together with GPEtn make up around 80% of the GPLs (Elabbadi, Ancelin, & Vial, 1997) and more recently it was found as enriched in the apicoplast (Botté et al., 2013). It has been shown that pathways for the synthesis of GPCho are functional in *Plasmodium*, one using GPSer and a second one via the decarboxylation of Serine to Ethanolamine, being the first one the preferred route in *P. falciparum* and *P. knowlesi*, at least (Elabbadi et al., 1997). This should explain the disappearance or depletion of GPSer molecules like GPSer(18:0/22:5), GPSer(20:4/18:0) and GPSer(18:0/22:6) from the non-infected

reticulocytes. The contrasting observations from the *Experiment 1* do not represent a contradiction as it corresponds to a single analysed sample (Figure 16 and S11).

Additionally, (Botté et al., 2013) also reported DG(32:0) as a molecule present in *P. falciparum* and being enriched in the apicoplast of this parasite species which here it was identified as DG(16:0/16:0), abundant in the infected samples but not detected in the non-infected cells. An alternative pathway for the production of DG is through the conversion of GPCho with the addition of Cer, this reaction will also lead to the production of SM (Mitamura & Palacpac, 2003) which could not be detected in this study.

These initial results are in agreement with the reported lipid composition of erythrocytes infected with *P. falciparum*, however the accumulation of certain lipids can be totally different when studying infected reticulocytes as this early erythrocyte is more metabolically active (Goh et al., 2007; E. Lee et al., 2014) and its own lipid metabolism can compensate or exacerbate the lipid changes product of the infecting parasite.

The difference in the host cell used for invasion in *P. vivax* and *P. falciparum* should implicate a different initial phospholipid material to be used by the parasite in case there is any variance. It is known that the lipid metabolism in the normocyte is almost non-functional (van Deenen & de Gier, 1974) so any change observed after the infection starts can be attributed to the parasite metabolism, this, however, is not the case when the young and metabolically active reticulocyte (Goh et al., 2007; E. Lee et al., 2014) is infected. This is why any well designed study should consider the generation of data from reticulocytes, normocytes and the same cell cultures now infected. For the presented master thesis, the correct comparison of the infected cord blood reticulocytes and a never infected cord blood reticulocyte cell culture was studied. This skips the common use of direct human blood extractions given the difficulty to get pure *P. vivax* cultures.

Understanding the contribution of each of the parts could help deciphering, for example, (i) what are the essential differences that confer different tropisms for each species and (ii) in giving new hints on how to maintain an *in vitro* culture of *P. vivax*. For example, a synthetic analogue of Choline (G25) has been shown to have an inhibitory activity for the *in vitro* growth of *P. falciparum* and *in vivo* activity of *P. falciparum* and *P. chabaudi* without adverse effects in mammalian cell cultures. Interestingly, this compound also showed activity against *P. vivax* and *P. cynomolgi in vitro* assays, both species can only invade reticulocytes indicating that there are no compensatory mechanisms in the metabolism of this host cell preventing the antimalarial activity of G25 (Wengelnik et al., 2002).

It is worth mentioning that the four identified lipids being overrepresented in the non-infected reticulocytes had a significant, although low, intensity in the *P. vivax*-infected samples. This situation suggests a slow degradative process by the parasite or an active process in the infected reticulocyte that maintains the phospholipids at detectable levels, probably because of an essential function for the membrane stability of the erythrocyte making their lost not permissible. However, this could also indicate an under identification in the non-infected samples.

One of the only two molecules that could not be detected in the non-infected sample but observed and identified in the infected one was the DG(16:0/16:0). Although this molecule has been detected and quantified in blood samples from healthy human adults (J. Lee et al., 2012), the full biosynthetic pathway is present in *P. falciparum* , which contains the Glycerol-3-phosphate O-acyltransferase (G3PAT or GAT, also named Acyl transferase I in humans) that catalyses the first acylation of Glycerol-3-phosphate to generate Lyso-PA. Then, 1-Acyl-sn-glycerol-3-phosphate O-acyltransferase (LPAAT, also named Acyl transferase II in humans) transfers a second acyl group to produce Phosphatidic acid, which is de-phosphorylated by Phosphatidate cytidylyltransferase (CDS) to produce DG.

As mentioned, there are two enzymes with acyltransferase activity with glycerol-3-phosphate as a substrate, they are G3PAT, encoded in the chromosome 13 of *P. falciparum* but functional in the Apicoplast (Lindner et al., 2014). The second enzyme is called GAT, encoded in chromosome 12 and functional in the Endoplasmic Reticulum (ER) membrane (Santiago, Zufferey, Mehra, Coleman, & Mamoun, 2004). The enzyme transferring the second acyl group is named LPAAT, found in the chromosome 14 of *P. falciparum* and active in the ER. G3PAT, GAT, LPAAT and CDS have all predicted homologues in *P. vivax* as reported in PlasmoDB, suggesting a conserved metabolism in both species.

For the case of GPCho(18:0/18:2), it has been detected and quantified in urine and it is predicted to be present in all tissues. Moreover, GPCho(36:2) has been reported as highly abundant in human blood plasma (Quehenberger et al., 2010) but no more details have been published about their presence in reticulocytes.

Galactolipids have been reported in *Toxoplasma gondii* and in *P. falciparum* (Marechal et al., 2002). Although LipidBlast computationally modelled spectra from more than 16,000 lipids corresponding to Mono- and di-galactosyldiacylglycerols as well as Sulfoquinovosyldiacylglycerols, none of the studied candidates produced a hit against these lipid classes.

The approach used to look for the presence of lipid classes by looking at particular ions that are specific for certain lipid classes was described before in a study of the lipids in rat retinal tissue (Busik et al., 2009). Although less accurate than the full spectra match, the results again showed a higher abundance of SM/GPCho lipids, as measured by the presence of phosphocholine and other molecules that can appear in the degradation of either SM or GPCho (see Table 5). However, dimethyl-ethanolaminephosphate, a marker for SM was never detected in *P. vivax* infected cell, suggesting that the SM/GPCho molecules were actually GPCho, confirming the enrichment observed in the identified lipids. The database used for Product Ion and Neutral Loss Scan was quite

reduced but the code that do the scanning from the MS/MS data showed its successful functionality at merging information obtained from more than one hit, different Scan modes and ionization modes. More ions have been described as signatures for lipid class identification but they lack resolution (See Table S1), as many of them are described by their nominal masses (when the masses of the atomic constituents are taken as integers), all these low-resolution cases were not considered for the presented results.

Lipidomics in Malaria Research

Although the sample number is relatively small, it represents the first study performed on short-term *in vitro* culture of *P. vivax* by using Cord Blood reticulocytes, an important initial step in the correct characterization of the *P. vivax* infection process.

Getting knowledge on the lipid composition could lead to different opportunities like: the inhibition of a new exclusive metabolic pathway in plasmodium, the identification of targets for the development of antibodies anti-lipids in a similar way as it was done for species-specific mycolic acids as an alternative to Tuberculosis diagnosis (Chan et al., 2013), to accelerate the diagnosis and reduce the misdiagnosis hence, improving the monitoring of malaria infections which is many times underestimated as it was revealed in a study of asymptomatic cases of *vivax* malaria in Cameroon (Fru-Cho et al., 2014) and in Malaysia with *P. knowlesi* (Yusof et al., 2014). And finally, and as a more ambitious perspective is that a *P. vivax*-specific lipid in the plasma membrane of the host reticulocyte could be used for the administration of a monoclonal antibody to modulate the opsonisation of the cell and promote its destruction through cytotoxic T-cells.

Bioinformatics

The programming language used is R (v.3.0.2), which is widely used and is commonly the preferred language by research groups working with biological data, in particular for high-throughput experiments. It allows a high degree of flexibility as it can be managed as a package with several independent functions that can take the data *in* or *out* at different points. The automation of the whole pipeline used here gives the warranty of the reproducibility of the results.

XCMS (Smith et al., 2006) has become broadly used and a proof of its success is the publication of several improvements to either the feature detection algorithm (Tautenhahn et al., 2008) or the online version (Gowda et al., 2014) which is user-friendly but less flexible to design *ad hoc* filtering and processing of the samples. For example, it does not allow (i) the exploration of normalization of the data, (ii) the elimination of represented features in a blank sample, (iii) the performance of statistical analysis based on technical replicates to eliminate non-reproducible features or (iv) further processing of data coming from two different experiments. These impediments could be overcome by producing a new pipeline that fulfils the requirements of the project. Moreover, the architecture of the designed pipeline can be useful for future characterization experiments from the *P. vivax* laboratory or other groups.

In many lipidomics studies the usage of QC samples are used to evaluate the quality, the signal-to-noise ratio and the improvement obtained after a normalization procedure was done (Berg et al., 2013; Veselkov et al., 2011). In cases where no standards are used, as it is the case of untargeted MS experiments, it has been suggested to use as a QC sample the mixture of both the sample and the control, this way in a multivariate analysis like PCA, the different QC samples should group together, otherwise their significant spread would indicate noise and the presence of artefacts. However, due to the nature of the present project, the control sample is a subset of the 'infected' sample, as only 10.9 to 15.1% of the cells were *P. vivax* positive (as shown in Table 4) making

it senseless the use of QC samples. Here, the use of the information obtained from technical replicates was used to measure the improvement of the normalization methods.

The evaluation of the normalization protocols is essential and must be done for every different experiment as the variations could come from different sources, as in this case that samples from *Experiment 1* and *Experiment 2* came from different protocols. In particular, for the aim of this project we are interested in relative changes and not in absolute levels. Lipidomics analyses do not give absolute numbers as it is the case of RNA sequencing data, however fold change can be assessed in a similar fashion as it has been done for RNA expression.

Cell count should always be the best way to avoid problems with normalization but it is not always achievable as in this case where the protocol used to enrich *P. vivax*-infected reticulocytes does not allow the cell counting. Furthermore, the use of parasitemia data is not meaningful here as non-infected samples are to be compared.

Between the methods tested for normalization, two very similar approaches based on median fold change normalization were used. Based on (Veselkov et al., 2011) it was assumed that given a large number of metabolites, the overall behaviour should be centered in a global median and only few will be outside, being those ones the interesting ones. For (Robinson & Oshlack, 2010), whose work was done in RNA-seq, they suggested to use the median obtained by samples instead, anticipating differences in several molecules but not in the more general distribution of the samples.

To choose the normalization method here, it was assumed that the majority of the features do not change (as the infected infected cells should retain housekeeping functions to allow the system to survive). However, we did not assume that the few ones that change do it in a conservative manner. Thus, we allowed the mean of the infected versus the not infected samples to change, selecting a normalization method

not based on median fold change. In other words, the preferred methods used on transcriptomics analyses were not selected based on a rational derived from the biological differences of the measured molecules.

Doing a normalization by protein estimation the results were not far dissimilar to the ones obtained by working with 'median fold by feature' which only re-scales all the values by a fixed number for the whole sample, and it is also very similar to 'total sum' normalization just adjusting the total to as standardized value (the sum of the intensities).

The lipidomics field face an additional problem that nucleic acids and proteins do not present: the absence of a reliable database for comparison of the obtained results. Current lipid databases are mainly filled with computational predictions which cannot contain all the possible options, hence leading to a number of missing assignations (Kind et al., 2013).

It has to be noted that for the processing of MS/MS data, a filtering of the spectra by using the retention time is necessary, otherwise several spectra will be present with retention times that will not match the initial retention time measured during the MS experiment. As this value should be constant because it depends on the lipid class as well as the size of the molecule and the length of the acyl groups, the initial value (obtained in the MS experiment) is used to narrow down the spectra of the MS/MS but a tolerance window is used (30 seconds) to allow expected variations, in particular because both experiments were performed at different times.

The reproducibility observed in the feature detection in the *Experiment 1* and *Experiment 2* was never below the 93% considering both ionization modes which indicates that the initial material had exceptional quality. Interestingly, less features were observed in the *Experiment 1*, that was expected as a different lipid extraction method was used, where the second allowed the purification of sterols in addition to

phospholipids (which were only detected in the first experiment). This difference in lipid extraction protocols could also explain why only 8 out of 13 identified lipids were found in the *Experiment 1* while the rest was not.

The presented code is also able to identify double charged lipids as it was the case of the feature in the positive mode of the *Experiment 1* (see Figure 11) with *m/z*= 646.4755 with additional peaks at *m/z*=645.9712 and one more at *m/z*= 645.4748 having differences of 0.5 as expected in a di-protonated molecule and in a 4:2:1 ratio. Unfortunately, no validation for this prediction was performed as the precursor corresponding to its monoisotope was not processed in Tandem MS.

Table 11 summarizes the different analyses that can be done with the presented package when compared to the mentioned programs XCMS and XCMS Online.

Table 11

	Developed package	**XCMS**	**XCMS Online**
Normalization options	✓ (Six algorithms)	×	×
Use of blank samples to identify noise	✓	NA	✓
Statistical tests based on technical replicates	✓	×	✓
Visualization of peaks	✓ (for any range)	✓ (Only default ranges)	✓
Identification of monoisotopic features	✓	×	×
Identification of double charged molecules	✓ (requires validation)	×	×
Inclusion of a lipid-specific library	✓ (LipidBlast)	NA	×
Visualizations	✓ (for intermediate steps and final results)	✓ (for global visualization)	✓ (Including interactive plots)

Table 11. Summary of functions that are implemented in the developed package and their status in XCMS and XCMS Online programs. NA, not applicable.

Conclusions

A robust and flexible R package was developed to fulfil all the requirements needed to identify ions being differentially abundant in a MS experiment and also to process the data generated by a Tandem MS experiment to identify reliable lipid species. This method can be applied to different data sets in an automatized and reproducible way.

The protocol was tested on human cord blood reticulocytes infected or not with *P. vivax*. A decrease in GPSer and increase in GPCho species are in agreement with the known metabolic pathways in *Plasmodium*, indicating an accurate data processing that can be applied, now, to a larger sample number.

References

Agnandji, S. T., Lell, B., Fernandes, J. F., Abossolo, B. P., Methogo, B. G., Kabwende, A. L., . . . Vansadia, P. (2012). A phase 3 trial of RTS,S/AS01 malaria vaccine in African infants. *N Engl J Med, 367(24)*(1533-4406 (Electronic)), 2284-2295.

Aikawa, M., Miller, L. H., & Rabbege, J. (1975). Caveola--vesicle complexes in the plasmalemma of erythrocytes infected by Plasmodium vivax and P cynomolgi. Unique structures related to Schuffner's dots. *Am J Pathol, 79(2)*(0002-9440 (Print)), 285-300. doi: D - NLM: PMC1912656 EDAT- 1975/05/01 MHDA- 1975/05/01 00:01 CRDT- 1975/05/01 00:00 PST - ppublish

Baird, J. K. (2013). Evidence and implications of mortality associated with acute Plasmodium vivax malaria. *Clin Microbiol Rev, 26(1)*(1098-6618 (Electronic)), 36-57.

Benning, C. (2008). A role for lipid trafficking in chloroplast biogenesis. *Prog Lipid Res., 47 (5)*(0163-7827 (Print)), 381-389.

Berg, M., Vanaerschot, M., Jankevics, A., Cuypers, B., Breitling, R., & Dujardin, J. C. (2013). LC-MS metabolomics from study design to data-analysis - using a versatile pathogen as a test case. *Comput Struct Biotechnol J, 4*, e201301002. doi: 10.5936/csbj.201301002

Berglund, M., & Wieser, M. E. (2011). Isotopic compositions of the elements 2009 (IUPAC Technical Report). *Pure Applied Chemistry, 83*(2), 397-410. doi: 10.1351/PAC-REP-10-06-02

Bolstad, B. M., Irizarry, R. A., Astrand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics, 19(2)*(1367-4803 (Print)), 185-193.

Botté, C. Y., Yamaryo-Botté, Y., Rupasinghe, T. W. T., Mullin, K. A., Macrae, J. I., Spurck, T. P., . . . McFadden, G. I. (2013). Atypical lipid composition in the purified relict plastid

(apicoplast) of malaria parasites. *Proc Natl Acad Sci U S A, 110*, 7506-7511.

Brugger, B., Erben, G., Sandhoff, R., Wieland, F. T., & Lehmann, W. D. (1997). Quantitative analysis of biological membrane lipids at the low picomole level by nano-electrospray ionization tandem mass spectrometry. *Proc Natl Acad Sci U S A, 94(6)*(0027-8424 (Print)), 2339-2344. doi: D - NLM: PMC20089 EDAT- 1997/03/18 MHDA- 1997/03/18 00:01 CRDT- 1997/03/18 00:00 PST - ppublish

Busik, J. V., Reid, G. E., & Lydic, T. A. (2009). Global analysis of retina lipids by complementary precursor ion and neutral loss mode tandem mass spectrometry. *Methods Mol Biol, 579*, 33-70. doi: 10.1007/978-1-60761-322-0_3

Chan, C. E., Zhao, B. Z., Cazenave-Gassiot, A., Pang, S. W., Bendt, A. K., Wenk, M. R., . . . Hanson, B. J. (2013). Novel phage display-derived mycolic acid-specific antibodies with potential for tuberculosis diagnosis. *J Lipid Res, 54(10)*(0022-2275 (Print)), 2924-2932.

DeLuca, D. S., Levin, J. Z., Sivachenko, A., Fennell, T., Nazaire, M. D., Williams, C., . . . Getz, G. (2012). RNA-SeQC: RNA-seq metrics for quality control and process optimization. *Bioinformatics, 28(11)*(1367-4811 (Electronic)), 1530-1532.

Dobson, G., Christie Ww Fau - Nikolova-Damyanova, B., & Nikolova-Damyanova, B. Silver ion chromatography of lipids and fatty acids. (1572-6495 (Print)).

Ejsing, C. S., Sampaio, J. L., Surendranath, V., Duchoslav, E., Ekroos, K., Klemm, R. W., . . . Shevchenko, A. (2009). Global analysis of the yeast lipidome by quantitative shotgun mass spectrometry. *Proc Natl Acad Sci U S A, 106(7)*(1091-6490 (Electronic)), 2136-2141.

Elabbadi, N., Ancelin, M. L., & Vial, H. J. (1997). Phospholipid metabolism of serine in Plasmodium-infected erythrocytes involves phosphatidylserine and direct serine decarboxylation. *Biochem J, 324*(0264-6021 (Print)).

Fahy, E., Sud, M., Cotter, D., Cotter, D., & Subramaniam, S. (2007). LIPID MAPS online tools for lipid research. *Nucleic Acids Res.*(1362-4962 (Electronic)). doi: D - NLM: PMC1933166 EDAT- 2007/06/23 09:00 MHDA- 2007/08/04 09:00 CRDT- 2007/06/23 09:00 PHST- 2007/06/21 [aheadofprint] AID - gkm324 [pii] AID - 10.1093/nar/gkm324 [doi] PST - ppublish

Fernando, H., Bhopale, K. K., Kondraganti, S., Kaphalia, B. S., & Shakeel Ansari, G. A. (2011). Lipidomic changes in rat liver after long-term exposure to ethanol. *Toxicol Appl Pharmacol, 255*(2), 127-137. doi: 10.1016/j.taap.2011.05.022

Fru-Cho, J., Bumah, V. V., Safeukui, I., Nkuo-Akenji, T., Titanji, V. P., & Haldar, K. (2014). Molecular typing reveals substantial Plasmodium vivax infection in asymptomatic adults in a rural area of Cameroon. *Malar J., 13:170*(1475-2875 (Electronic)).

Galinski, M. R., Meyer, E. V., & Barnwell, J. W. (2013). Plasmodium vivax: modern strategies to study a persistent parasite's life cycle. *Adv Parasitol, 81*, 1-26. doi: 10.1016/B978-0-12-407826-0.00001-1

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., . . . Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol., 5(10)*(1465-6914 (Electronic)).

Genton, B., D'Acremont, V., Rare, L., Baea, K., Reeder, J. C., Alpers, M. P., & Muller, I. (2008). Plasmodium vivax and mixed infections are associated with severe malaria in children: a prospective cohort study from Papua New Guinea. *PLoS Med, 5(6)*(1549-1676 (Electronic)). doi: D - NLM: PMC2429951 EDAT- 2008/06/20 09:00 MHDA- 2008/08/08 09:00 CRDT- 2008/06/20 09:00 PHST- 2007/12/07 [received] PHST- 2008/05/02 [accepted] AID - 07-PLME-RA-2233 [pii] AID - 10.1371/journal.pmed.0050127 [doi] PST - ppublish

Gething, P. W., I.R., E., Moyes, C. L., Smith, D. L., Battle, K. E., Guerra, C. A., . . . Hay, S. I. (2012). A long neglected world malaria map: Plasmodium vivax endemicity in 2010. *PLoS Negl Trop Dis., 6(9)*(1935-2735 (Electronic)). doi: D - NLM: PMC3435256 EDAT- 2012/09/13 06:00 MHDA- 2013/01/29 06:00 CRDT- 2012/09/13 06:00 PHST- 2012/04/24 [received] PHST- 2012/07/29 [accepted] PHST- 2012/09/06 [epublish] AID - 10.1371/journal.pntd.0001814 [doi] AID - PNTD-D-12-00489 [pii] PST - ppublish

Goh, S. H., Josleyn, M., Lee, Y. T., Danner, R. L., Gherman, R. B., Cam, M. C., & Miller, J. L. (2007). The human reticulocyte transcriptome. *Physiol Genomics, 30*(2), 172-178. doi: 10.1152/physiolgenomics.00247.2006

Gowda, H., Ivanisevic, J., Johnson, C. H., Kurczy, M. E., Benton, H. P., Rinehart, D., . . . Siuzdak, G. (2014). Interactive XCMS Online: simplifying advanced metabolomic data processing and subsequent statistical analyses. *Anal Chem., 86(14)*(1520-6882 (Electronic)), 6931-6939.

Han, X., & Gross, R. W. (1995). Structural determination of picomole amounts of phospholipids via electrospray ionization tandem mass spectrometry. *J Am Soc Mass Spectrom., 6(12)*(1044-0305 (Print)), 1202-1210.

Han, X., & Gross, R. W. (2003). Global analyses of cellular lipidomes directly from crude extracts of biological samples by ESI mass spectrometry: a bridge to lipidomics. *J Lipid Res, 44*(6), 1071-1079. doi: 10.1194/jlr.R300004-JLR200

Han, X., & Gross, R. W. (2005). Shotgun lipidomics: electrospray ionization mass spectrometric analysis and quantitation of cellular lipidomes directly from crude extracts of biological samples. *Mass Spectrom Rev, 24*(3), 367-412. doi: 10.1002/mas.20023

Han, X., Yang, K., Yang, J., Cheng, H., & Gross, R. W. (2006). Shotgun lipidomics of cardiolipin molecular species in lipid extracts of biological samples. *J Lipid Res, 47*(4), 864-879. doi: 10.1194/jlr.D500044-JLR200

Handayani, S., Chiu, D. T., Tjitra, E., Kuo, J. S., Lampah, D., Kenangalem, E., . . . Russell, B. (2009). High deformability of Plasmodium vivax-infected red blood cells under microfluidic conditions. *J Infect Dis, 199(3)*(0022-1899 (Print)), 445-450.

Hoffman, E., & Stroobant, V. (2007). *Mass Spectrometry Principles and Applications* (Wiley Ed. Third ed.).

Holz, G. G. (1977). Lipids and the malarial parasite. *Bulletin of the World Health Organization, 55*(2-3), 237-248.

Ishizuka, I. (1997). Chemistry and functional distribution of sulfoglycolipids. *Progress in Lipid Research, 36*, 245-319. doi: 10.1016/S0163-7827(97)00011-8

Kind, T., Liu, K. H., Lee do, Y., DeFelice, B., Meissen, J. K., & Fiehn, O. (2013). LipidBlast in silico tandem mass spectrometry database for lipid identification. *Nat Methods, 10*(8), 755-758. doi: 10.1038/nmeth.2551

Kitchen, S. F. (1938). The infection of reticulocytes by *Plasmodium vivax*. *American Journal of Tropical Medicine, 18*, 347-359.

Kwon, Y. K., Ha, I. J., Bae, H. W., Jang, W. G., Yun, H. J., Kim, S. R., . . . Hwang, G. S. (2014). Dose-dependent metabolic alterations in human cells exposed to gamma irradiation. *PLoS One, 9(11)*(1932-6203 (Electronic)).

Layre, E., & Moody, D. B. (2013). Lipidomic profiling of model organisms and the world's major pathogens. *Biochimie, 95*(1), 109-115. doi: 10.1016/j.biochi.2012.08.012

Lee, E., Choi, H. S., Hwang, J. H., Hoh, J. K., Cho, Y. H., & Baek, E. J. (2014). The RNA in reticulocytes is not just debris: it is necessary for the final stages of erythrocyte formation. *Blood Cells Mol Dis, 53*(1-2), 1-10. doi: 10.1016/j.bcmd.2014.02.009

Lee, J., Park J Fau - Lim, M.-s., Lim Ms Fau - Seong, S. J., Seong Sj Fau - Seo, J. J., Seo Jj Fau - Park, S. M., Park Sm Fau - Lee, H. W., . . . Yoon, Y. R. (2012). Quantile normalization approach for liquid chromatography-mass spectrometry-based metabolomic data from healthy human volunteers. (1348-2246 (Electronic)).

Li, M., Yang, L., Bai, Y., & Liu, H. (2014). Analytical methods in lipidomics and their applications. *Anal Chem, 86*(1), 161-175. doi: 10.1021/ac403554h

Lindner, S. E., Sartain, M. J., Hayes, K., Harupa, A., Moritz, R. L., Kappe, S. H., & Vaughan, A. M. (2014). Enzymes involved in plastid-targeted phosphatidic acid synthesis are essential for Plasmodium yoelii liver-stage development. *Mol Microbiol, 91(4)*(1365-2958 (Electronic)).

Lingelbach, K., & Joiner, K. A. (1998). The parasitophorous vacuole membrane surrounding Plasmodium and Toxoplasma: an unusual compartment in infected cells. *J Cell Sci, 111*, 1467-1475.

Malleret, B., Li, A., Zhang, R., Tan, K. S., Suwanarusk, R., Claser, C., . . . Russell, B. (2014). Plasmodium vivax: restricted tropism and rapid remodelling of CD71

positive reticulocytes. LID - blood-2014-08-596015 [pii]. *Blood*(1528-0020 (Electronic)).

Malleret, B., Xu, F., Mohandas, N., Suwanarusk, R., Chu, C. S., Leite, J. A., . . . Russell, B. (2013). Significant biochemical, biophysical and metabolic diversity in circulating human cord blood reticulocytes. *PLoS One, 8(10)*(1932-6203 (Electronic)).

Marechal, E., Azzouz, N., Santos de Macedo, C., Block, M. A., Feagin, J. E., Schwarz, R. T., & Joyard, J. (2002). Synthesis of Chloroplast Galactolipids in Apicomplexan Parasites. *Eukaryotic Cell, 1*(4), 653-656. doi: 10.1128/ec.1.4.653-656.2002

Mitamura, T., & Palacpac, N. M. Q. (2003). Lipid metabolism in Plasmodium falciparum-infected erythrocytes: possible new targets for malaria chemotherapy. *Microbes and Infection, 5*(6), 545-552. doi: 10.1016/s1286-4579(03)00070-4

Mueller, I., Galinski, M. R., Baird, J. K., Carlton, J. M., Kochar, D. K., Alonso, P. L., & del Portillo, H. A. (2009). Key gaps in the knowledge of Plasmodium vivax, a neglected human malaria parasite. *Lancet Infect Dis, 9(9)*, 555-566.

Narayanaswamy, P., Shinde, S., Sulc, R., Kraut, R., Staples, G., Thiam, C. H., . . . Wenk, M. R. (2014). Lipidomic "deep profiling": an enhanced workflow to reveal new molecular species of signaling lipids. *Anal Chem, 86*(6), 3043-3047. doi: 10.1021/ac4039652

Oursel, D., Loutelier-Bourhis, C., Orange, N., Chevalier, S., Norris, V., & Lange, C. M. (2007). Lipid composition of membranes of Escherichia coli by liquid chromatography/tandem mass spectrometry using negative electrospray ionization. *Rapid Commun Mass Spectrom, 21*(11), 1721-1728. doi: 10.1002/rcm.3013

Quehenberger, O., Armando, A. M., Brown, A. H., Milne, S. B., Myers, D. S., Merrill, A. H., . . . Dennis, E. A. (2010). Lipidomics reveals a remarkable diversity of lipids in human plasma. *J Lipid Res, 51*(11), 3299-3305. doi: 10.1194/jlr.M009449

Quispe, A. M., Pozo, E., Guerrero, E., Durand, S., Baldeviano, G. C., Edgel, K. A., . . . Lescano, A. G. (2014). Plasmodium vivax hospitalizations in a monoendemic malaria region: severe vivax malaria? *Am J Trop Med Hyg, 91(1)*(1476-1645 (Electronic)), 11-17.

Report, W. M. (2013). World Health Organization.

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol, 11*(3), R25. doi: 10.1186/gb-2010-11-3-r25

Samuelsson, B., & Samuelsson, K. (1960). Gas-liquid chromatography-mass spectrometry of synthetic ceramides. *J Lipid Res, 10*(0022-2275 (Print)), 41-46.

Santiago, T. C., Zufferey, R., Mehra, R. S., Coleman, R. A., & Mamoun, C. B. (2004). The Plasmodium falciparum PfGatp is an endoplasmic reticulum membrane protein important for the initial step of malarial glycerolipid synthesis. *J Biol Chem, 279(10)*(0021-9258 (Print)), 9222-9232.

Shi, L., Reid, L. H., Jones, W. D., Shippy, R., Warrington, J. A., Baker, S. C., . . . Kawasaki, E. S. (2006). The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol., 24(9)*(1087-0156 (Print)), 1151-1161.

Siddiqui, W. A., Schnell, J. V., & Geiman, Q. M. (1970). In vitro cultivation of Plasmodium falciparum. *Am J Trop Med Hyg, 19(4)*(0002-9637 (Print)), 586-591.

Singh, A., & Prasad, R. (2011). Comparative lipidomics of azole sensitive and resistant clinical isolates of Candida albicans reveals unexpected diversity in molecular lipid imprints. *PLoS One, 6*(4), e19266. doi: 10.1371/journal.pone.0019266

Smith, C. A., Want, E. J., O'Maille, G., Abagyan, R., & Siuzdak, G. (2006). XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem., 78(3)*(0003-2700 (Print)), 779-787.

Smyth, G. K. (2005). Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry & W. Huber (Eds.), *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. New York: Springer.

Stein, S. E., & Scott, D. R. (1994). Optimization and testing of mass spectral library search algorithms for compound identification. *J Am Soc Mass Spectrom., 5(9)*(1044-0305 (Print)), 859-866. doi: 10.1016/1044-0305(94)87009-8

Surolia, N., & Surolia, A. (2001). Triclosan offers protection against blood stages of malaria by inhibiting enoyl-ACP reductase of Plasmodium falciparum. *Nat Med, 7(2)*(1078-8956 (Print)), 167-173.

Suwanarusk, R., Cooke, B. M., Dondorp, A. M., Silamut, K., Sattabongkot, J., White, N. J., & Udomsangpetch, R. (2004). The deformability of red blood cells parasitized by Plasmodium falciparum and P. vivax. *J Infect Dis, 189(2)*(0022-1899 (Print)), 190-194.

Ta, T. H., Hisam, S., Lanza, M., Jiram, A. I., Ismail, N., & Rubio, J. M. (2014). First case of a naturally acquired human infection with Plasmodium cynomolgi. *Malar J, 13:68*(1475-2875 (Electronic)).

Tautenhahn, R., Bottcher, C., & Neumann, S. (2008). Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics*(1471-2105 (Electronic)).

van Deenen, L. L. M., & de Gier, J. (1974). *Lipids of the red cell membrane*. New York: Academic Press.

van Dooren, G. G., & Striepen, B. (2013). The algal past and parasite present of the apicoplast. *Annu Rev Microbiol, 67*(1545-3251 (Electronic)), 271-289.

Veselkov, K. A., Vingara, L. K., Masson, P., Robinette, S. L., Want, E., Li, J. V., . . . Nicholson, J. K. (2011). Optimized preprocessing of ultra-performance liquid chromatography/mass spectrometry urinary metabolic profiles for improved information recovery. *Anal Chem, 83*(15), 5864-5872. doi: 10.1021/ac201065j

Waller, R. F., Keeling, P. J., Donald, R. G., Striepen, B., Handman, E., Lang-Unnasch, N., . . . McFadden, G. I. (1998). Nuclear-encoded proteins target to the plastid in Toxoplasma gondii and Plasmodium falciparum. *Proc Natl Acad Sci U S A, 95(21)*(0027-8424 (Print)), 12352-12357. doi: D - NLM: PMC22835 EDAT- 1998/10/15 MHDA- 1998/10/15 00:01 CRDT- 1998/10/15 00:00 PST - ppublish

Welti, R., Mui, E., Sparks, A., Wernimont, S., Isaac, G., Kirisits, M., . . . McLeod, R. (2007). Lipidomic analysis of Toxoplasma gondii reveals unusual polar lipids. *Biochemistry, 46*(48), 13882-13890. doi: 10.1021/bi7011993

Welti, R., & Wang, X. (2004). Lipid species profiling: a high-throughput approach to identify lipid compositional changes and determine the function of genes

involved in lipid metabolism and signaling. *Curr Opin Plant Biol, 7*(3), 337-344. doi: 10.1016/j.pbi.2004.03.011

Wengelnik, K., Vidal, V., Ancelin, M. L., Cathiard, A. M., Morgat, J. L., Kocken, C. H., . . . Vial, H. J. (2002). A class of potent antimalarials and their specific accumulation in infected erythrocytes. *Science, 295*(1095-9203 (Electronic)), 1311-1314.

Wenk, M. R. (2005). The emerging field of lipidomics. *Nat Rev Drug Discov, 4*(7), 594-610. doi: 10.1038/nrd1776

Yusof, R., Lau, Y. L., Mahmud, R., Fong, M. Y., Jelip, J., Ngian, H. U., . . . Mohd Ali, M. (2014). High proportion of knowlesi malaria in recent malaria cases in Malaysia. *Malar J, 13:168*(1475-2875 (Electronic)).

Appendix A

Supplementary figures.

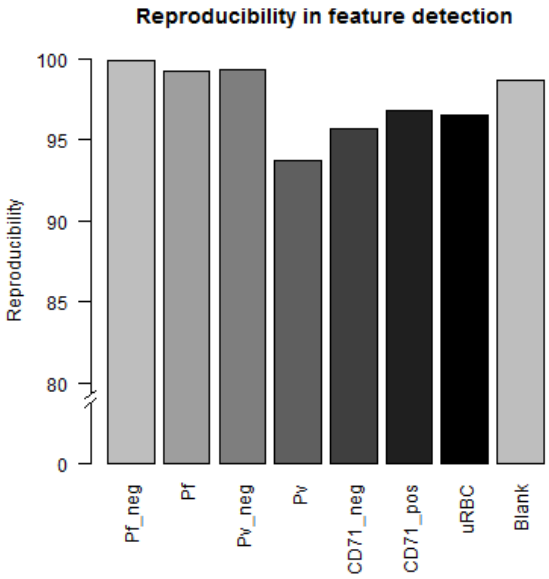


Figure S1. Reproducibility in feature detection in the first experiment in positive mode. The order in the X-axis is the same as in Table 1.

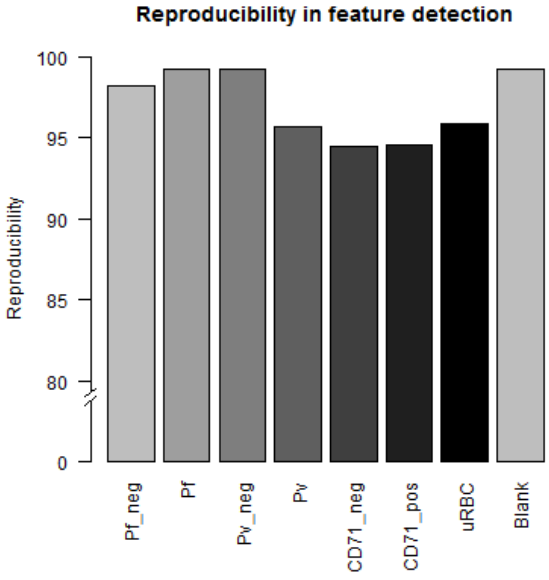


Figure S2. Reproducibility in feature detection in the first experiment in negative mode. The order in the X-axis is the same as in Table 1.

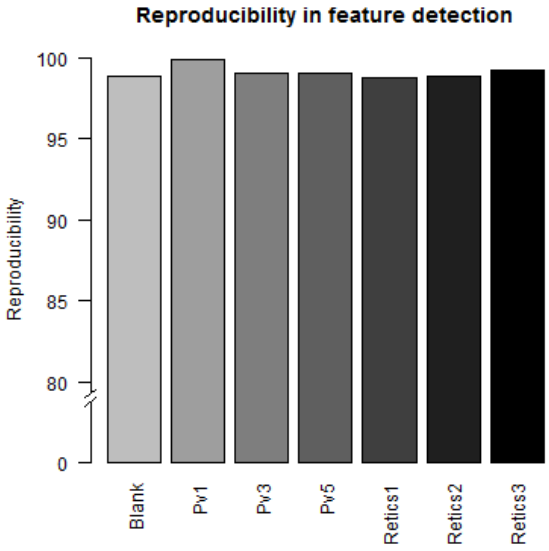


Figure S3. Reproducibility in feature detection in the second experiment in negative mode.

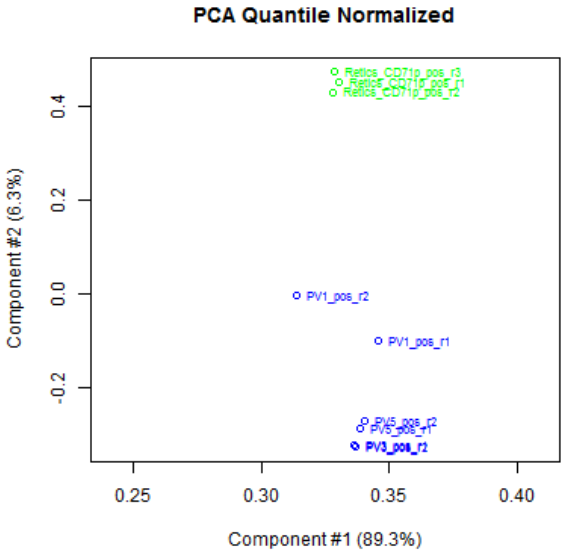


Figure S4. Principal component analysis for quantile normalization method using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.

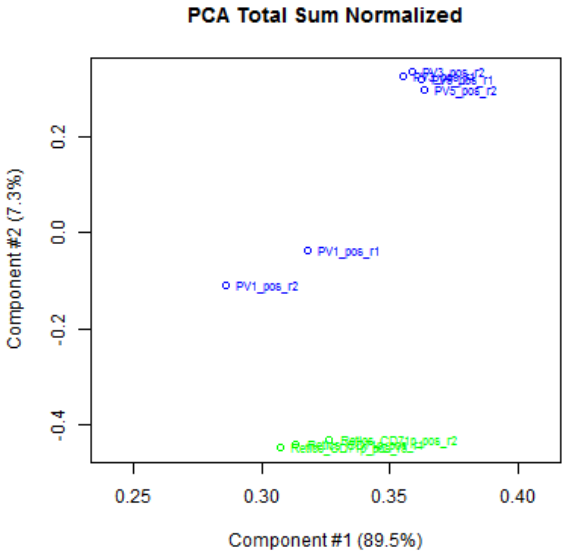
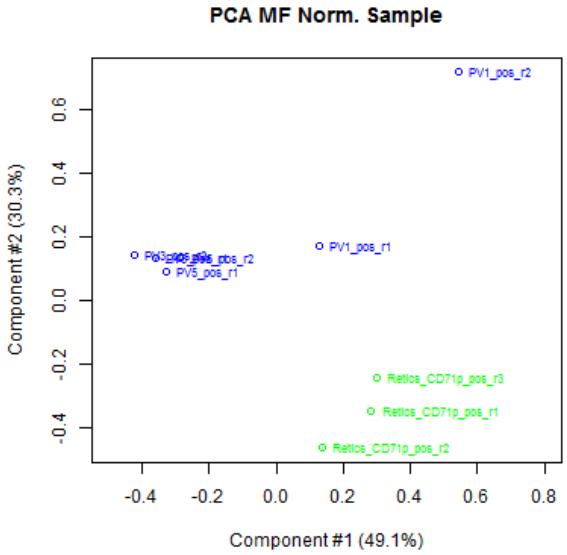


Figure S5. Principal component analysis for total sum normalization method using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.



8

Figure S6. Principal component analysis for median fold normalization method by sample using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.

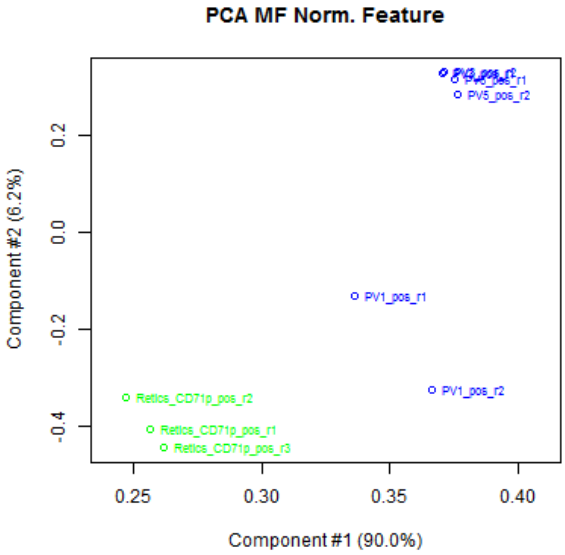


Figure S7. Principal component analysis for median fold normalization method by feature using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.

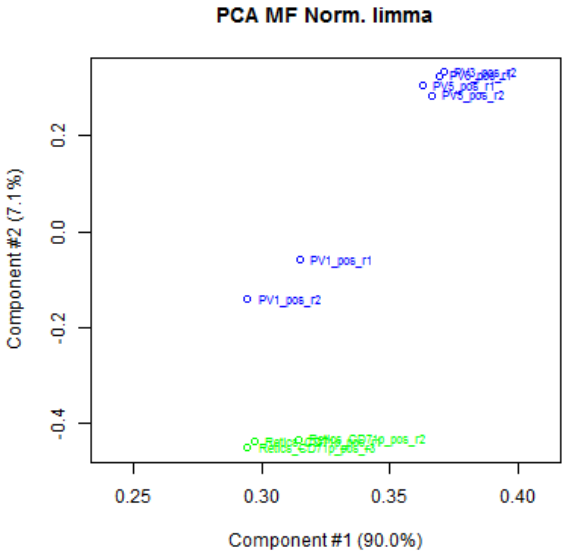


Figure S8. Principal component analysis for median fold normalization method following the function in the 'limma' package form Bioconductor (Smyth, 2005) and using the data for the positive mode. X-axis, principal component; Y-axis, second component. Green, non-infected samples; blue, infected samples.

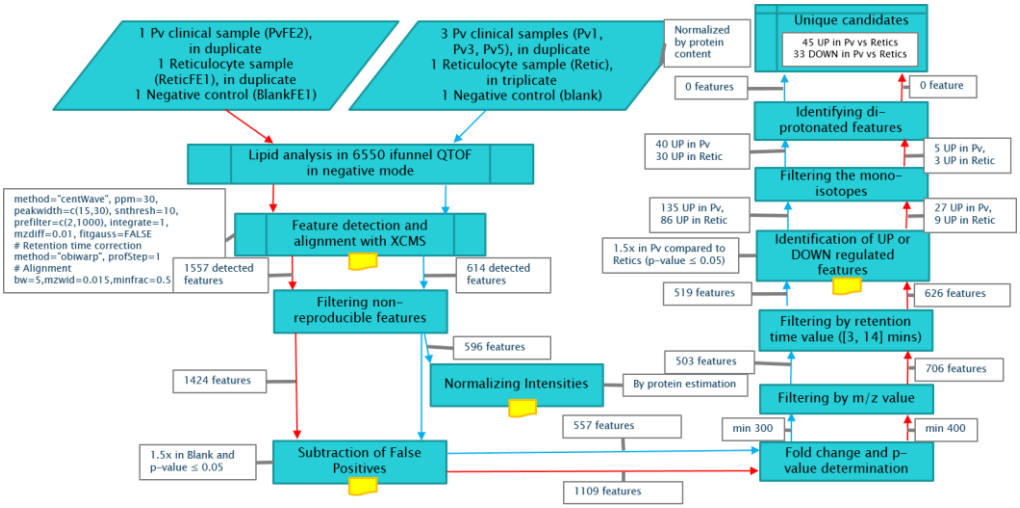


Figure S9. Flux diagram including the number of filtered features in each step in the negative mode.

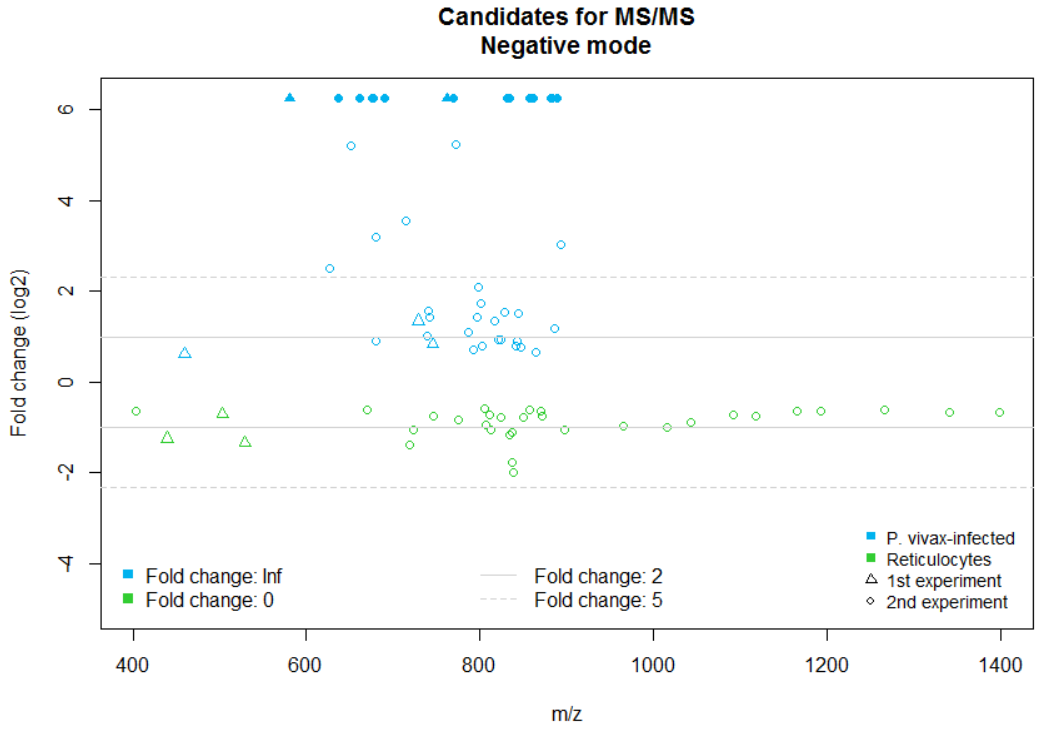


Figure S10. Distribution of the candidates for MS/MS in negative mode. The solid blue symbols denote an infinite fold change obtained when no signal existed in the non-infected sample.

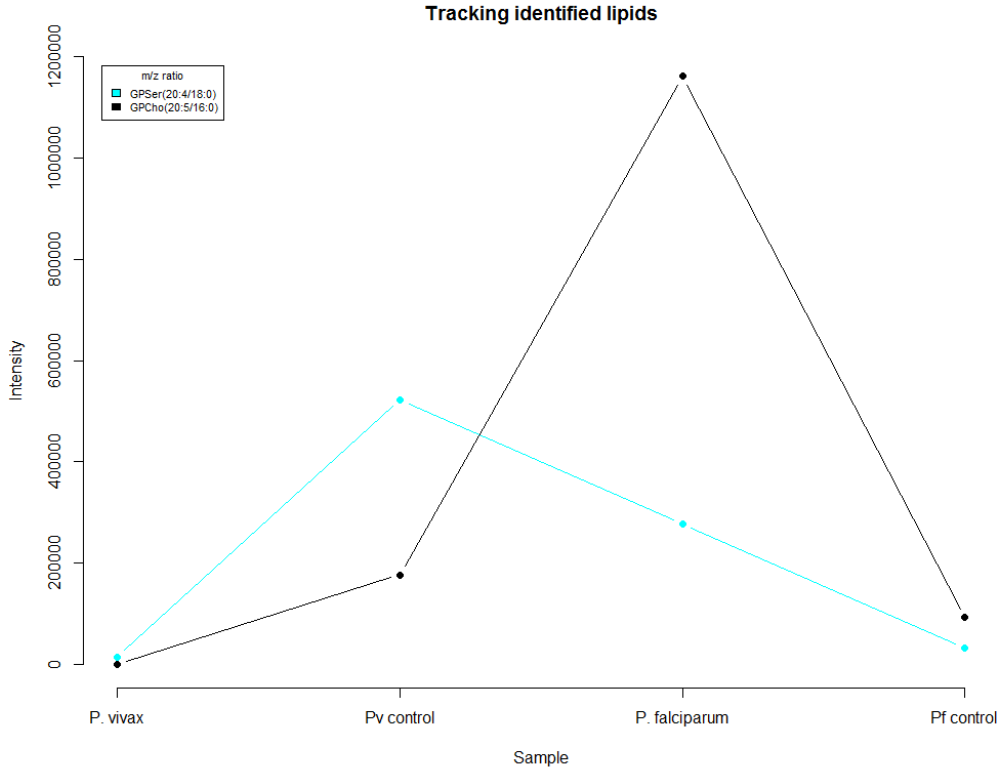


Figure S11. Tracking of the identified lipids being over-represented in the *P. vivax*-infected samples as found in the *Experiment 2* in the data from *Experiment 1*. Any precursor can be tracked to observe its level of abundance in a different experiment. Different behaviours can be identified. In the X-axis, the samples are as follows: *P. vivax*-infected reticulocytes, non-infected reticulocytes, *P. falciparum*-infected erythrocytes, non-infected erythrocytes. GPSer, glycerophosphatidylserine; GPCho, glycerophosphatidylcholine.

Supplementary Tables

Table S1

Lipid Class	Mode	Precursor Ion	MS/MS Scan Type	Molecule	Exact Mass	Reference
SM, GPCho	Neg	[M+Cl]-	NLS	CH3Cl	50.488	Busik2009
SM, GPCho	Neg	[M+CH3OCO2]-	NLS	CH3OCO2H+(CH3)3N	135.1617	Busik2009
SM, GPCho	Neg	[M+CH3OCO2]-	NLS	CH3OCO2+(CH3)3NCHCH2	161.1989	Busik2009
SM, GPCho	Pos	[M+H]+	PIS	Phosphocholine	184.150662	Han2005
SM, GPCho	Pos	[M+Na]+	PIS	Sodium cyclophosphane	147.022	Han1995
GPCho	Pos	[M+Na]+	NLS	Sodium cholinephosphate	205.1404313	Han2005
SM, GPCho	Pos	[M+Na]+	NLS	Neutral phosphocholine	183.150662	Brugger1997
GPEtn	Pos	[M+H]+, M+Na]+	NLS	Phosphoethanolamine	141.063	Brugger1997
GPEtn	Neg	[M-H]-	PIS	Glycerol phosphoethanolamine derivative	196.0275	Han2005
GPIns	Neg	[M-H]-	PIS	Dehydrated phosphoinositol	241.104602	Han2005
GPIns	Pos	[M+NH4]+	NLS	Phosphoinositol+NH3	277.166282	Busik2009
GPSer	Pos	[M+H]+, [M+Na]+	NLS	Phosphoserine	185.072	Busik2009
GPSer	Pos	[M+Na]+	PIS	Sodium phosphoserine	208.0617693	Busik2009
GPSer	Neg	[M-H]-	NLS	Serine-H2O	87.0777	Han2005
CE	Pos	[M+NH4]+	PIS	Cholestane cation	369.3	Han2005
MG, DG	Pos	[M+NH4]+	NLS	H2O+NH3	35.0458	Busik2009
GPGro	Pos	[M+NH4]+	NLS	Phosphoglycerol+NH3	189.104222	Busik2009
18:0-based ceramides	Pos	[M+H]+	PIS	Sphinganine derivative	266	Busik2009
18:1-based ceramides	Pos	[M+H]+	PIS	Sphingosine derivative	264	Busik2009
18:2-based ceramides	Pos	[M+H]+	PIS	Sphingadienine derivative	262	Busik2009
18:0-based ceramides	Neg	[M-H]-	NLS	16:0 aldehyde	258	Busik2009
18:1-based ceramides	Neg	[M-H]-	NLS	16:1 aldehyde	256	Busik2009
18:2-based ceramides	Neg	[M-H]-	NLS	16:2 aldehyde	254	Busik2009
PL	Neg	[M-H]-	NLS	Stearic acid (18:0)	284.2715	Brugger1997
PL	Neg	[M-H]-	NLS	Oleic acid (18:1)	282.2559	Brugger1997
PL	Neg	[M-H]-	NLS	Linoleic acid (18:2)	280.2402	Brugger1997
GPA	Pos	[M+NH4]+	NLS	NH3+H3PO4	115.025702	Busik2009
GPGro, GPA	Neg	[M-H]-	PIS	Cyclic glycerophosphate derivative	153.058442	Han2005
GPIns	Neg	[M-H]-	NLS	Inositol unit -H2O	162.14058	Brugger1997
GPIns	Neg	[M-H]-	PIS	[H2PO4]-	96.9872	Brugger1997
PL	Neg	[M-H]-	PIS	Inositol unit -H2O	162.14058	Brugger1997
PL	Neg	[M-H]-	PIS	[H2PO4]-	96.9872	Brugger1997
SM	Neg	[M-H]-	PIS	Dimethyl-ethanolaminephosphate	168.108202	Brugger1997
PL	Neg	[M-H]-	PIS	14:0 fatty acid anion	227.2017	-
MGDG, DGDG	Pos	[M+Li(Na)]+	PIS	Li(Na)+galactose derivative	227	Han2005
SQDG	Neg	[M-H]-	PIS	Galactose derivative	225	Han2005

Table S5. Exact mass values of Product Ion and Neutral Loss Scan experiments. SM, Sphingomyelin; GPCho, Glycerophosphatidylcholine; GPEtn, Glycerophosphatidylethanolamine; GPIns, Glycerophosphatidylinositol; GPSer, Glycerophosphatidylserine; CE, Cholesteryl ester; MG, Monoacylglycerol; DG, Diacylglycerol; GPGro, Glycerophosphatidylglycerol; GPA, Glycerophosphatidic acid; PL, Phospholipid; MGDG, Monogalactosyldiacylglycerol; DGDG, Digalactosyldiacylglycerol; SQDG, Sulfoquinovosyldiacylglycerol; Pos, Positive ionization mode; Neg, Negative ionization mode; NLS, Neutral Loss Scan; PIS, Parent Ion Scan; Busik2009, (Busik et al., 2009); Han2005, (Han & Gross, 2005).

Table S2.

Precursor	Possible Lipid Class	Support*	Secondary identification
439.2818909	SM	1	-
604.3875732	MG, DG	1	-
804.5751343	SM, GPCho	1	-
806.5330200	GPIns	1	-
810.5297852	GPSer	1	-
826.5799561	MG, DG	1	-
827.7203979	MG, DG	2	-
834.5294189	GPSer	2	-
836.5441895	GPSer	1	-
836.5783081	GPSer	5	-
837.6919556	CE	1	-
842.6392212	GPSer	3	-
897.7047729	SM	2	-
1024.6757813	SM, GPCho	1	-
1042.2441406	SM, GPCho	1	-
1091.2396240	GPEtn	1	-
1397.8868408	GPSer	1	-

Table S2. Precursors from the over represented ions in the non-infected reticulocytes sample with an assigned Lipid Class. The asterisk indicates the number of precursors with same *m/z* but different ionization mode or retention time that contained an ion corresponding with a Product Ion or the Neutral Loss that supported the assigned Lipid Class. When the same precursor was assigned to more than one Product Ion or Neutral Loss corresponding to a different Lipid Class the second assigned class is reported. SM, Sphingomyelin; GPCho, Glycerophosphatidylcholine; GPEtn, Glycerophosphatidylethanolamine; GPIns, Glycerophosphatidylinositol; GPSer, Glycerophosphatidylserine; CE, Cholesteryl ester; MG, Monoacylglycerol; DG, Diacylglycerol; GPGro, Glycerophosphatidylglycerol; GPA, Glycerophosphatidic acid.

Table S3.

Precursor	Possible Lipid Class	Support*	Secondary identification
610.5592041	MG, DG	2	-
612.5740356	MG, DG	4	-
625.5357056	SM, GPCho	1	-
634.5590210	MG, DG	1	-
638.5890503	MG, DG	2	-
662.5881958	MG, DG	2	-
692.5393677	GPA	1	SM, GPCho
701.5739746	SM, GPCho	1	-
703.5913086	SM, GPCho	1	-
716.5391846	GPA	1	-
718.5551147	GPA	1	-
718.5886841	GPEtn	3	-
720.6036987	GPEtn	1	SM, GPCho
734.5853271	SM, GPCho	1	-
738.5087891	GPEtn	1	-
742.5795898	GPEtn	3	-
744.5648193	GPEtn	1	SM, GPCho
744.5753174	GPEtn	2	-
758.5848999	SM, GPCho	1	-
762.6159668	SM, GPCho	5	-
766.5841675	GPEtn	6	SM, GPCho
774.6490479	GPEtn	2	-
786.5147705	GPEtn	1	-
786.6156006	SM, GPCho	1	-
788.5430298	GPEtn	1	-
788.6213379	GPSer	1	SM, GPCho
788.6354980	SM, GPCho	5	-
794.6160278	GPEtn	4	SM, GPCho
796.6001587	SM, GPCho	2	-
806.5835571	SM, GPCho	2	-
810.6148071	SM, GPCho	4	-
812.6311646	SM, GPCho	3	-
815.5678101	GPEtn	2	-
820.6064453	GPSer	1	-
824.5780029	GPGro, GPA	1	-
830.5809937	SM, GPCho	1	-
834.6127319	SM, GPCho	1	-
836.6265259	SM, GPCho	1	-
844.6638794	GPSer	1	-
859.5316772	GPGro, GPA	1	-
861.5485840	GPIns	1	-

Table S3. Precursors from the over represented ions in the *P. vivax*-infected reticulocytes sample with an assigned Lipid Class. The asterisk indicates the number of precursors with same *m/z* but different ionization mode or retention time that contained an ion corresponding with a Product Ion or the Neutral Loss that supported the assigned Lipid Class. When the same precursor was assigned to more than one Product Ion or Neutral Loss corresponding to a different Lipid Class the second assigned class is reported. SM, Sphingomyelin; GPCho, Glycerophosphatidylcholine; GPEtn, Glycerophosphatidylethanolamine; GPIns, Glycerophosphatidylinositol; GPSer, Glycerophosphatidylserine; CE, Cholesteryl ester; MG, Monoacylglycerol; DG, Diacylglycerol; GPGro, Glycerophosphatidylglycerol; GPA, Glycerophosphatidic acid.

Appendix B

Complete code of the developed R package used to analyse the data

filterTechReps.R

```
# To filter the technical replicates. The indicated proportion of the replicates must be above or
under
# the specified threshold.
# The following function does not need the filterIntensity() filter.
# The function must be run per each sample.
## USAGE: todosPfyControls<-filterTechReps(todos, 9:10)
filterTechReps<-function(tabla, replicates, intensity.cutoff=5000, techRep.cutoff=0.66){  #
Must be used 0.66 because 0.67*3=2.01
        vector<-apply(tabla, 1, function(linea){
                # First case: replicates have an intensity above the cut off.

        if(sum(linea[replicates]>=intensity.cutoff)>techRep.cutoff*length(replicates))
                        return(TRUE)
                # Second case: replicates have an intensity under the cut off.
                else
if(sum(linea[replicates]<intensity.cutoff)>techRep.cutoff*length(replicates))
                        return(TRUE)
                # Third case: replicates are inconsistent on being above or under the cutoff.
These features
                # will be eliminated.
                else{
                        return(FALSE)
                }
        })
        return(tabla[which(vector),])
}
```

featureNormalization.R

```
# This fuction normalize the Feature Intensities from an xset comming either from xcmsSet(),
retcor(),
# groups() or fillPeaks(), (an xcmsSet object).
# USAGE: normalized<-featureNormalization(xset3, method="quantile",
blank.columns=c(9:10))
featureNormalization<-function(xset=xset, method="quantile", blank.columns=NULL){
 xsetTable<-peakTable(xset)
 classinfo<-attributes(xset)[5][[1]]
 unique_classes<-unique(classinfo)
 num_classes<-dim(unique_classes)[1]
 columns<-as.numeric(7+num_classes)
 columns1<-columns+1
 meta<-data.frame(xsetTable[,c(1:columns)])
 # Columns corresponding to Blank samples are not considered for the normalization.
 intensity.columns<-setdiff(c(columns1:ncol(xsetTable)), blank.columns)
 originalIntensities<-xsetTable[,intensity.columns]
 if(method=="quantile"){
   QnormalizedIntensities<-normalizeQuantiles(originalIntensities)
#    colnames(QnormalizedIntensities)<-colnames(originalIntensities)
   QnormalizedIntensities<-data.frame(QnormalizedIntensities)
#    output<-cbind(meta, QnormalizedIntensities)
   normalizedIntensities<-QnormalizedIntensities
 }
 else if(method=="totalsum"){
```

```
   normalizedIntensities<-apply(originalIntensities, 2, function(sample){
     return(sample/sum(sample))
   })
   normalizedIntensities<-data.frame(normalizedIntensities)
#    output<-cbind(meta, normalizedIntensities)
#    normalizedIntensities<-t(apply(normalizedIntensities, 1, unlist))
  }
  else if(method=="medianfold.sample"){
   # This method centers each distribution in the median of the fold changes.
   # Method based on LB idea, obtaining the median values per sample.
   # Replacing ceros for 0.0001 to avoid NaN or Inf values.
   originalIntensities<-apply(originalIntensities, 2, function(x){
     x[which(x==0)]<-1
     return(x)
   })
   average<-apply(t(originalIntensities), 1, median) # Changed to median NOT as suggested by
LB
   foldpersample<-sapply(1:dim(originalIntensities)[2], function(index){
     return(originalIntensities[,index]/average[index])
   })
   mfnormalizedIntensities<-t(sapply(1:dim(originalIntensities)[1], function(index){
     return(originalIntensities[index,]/mean(foldpersample[index,])) # Changed to mean NOT
as suggested by LB
   }))
   colnames(mfnormalizedIntensities)<-colnames(originalIntensities)
   mfnormalizedIntensities<-data.frame(mfnormalizedIntensities)
#    output<-cbind(meta, mfnormalizedIntensities)
   normalizedIntensities<-mfnormalizedIntensities
  }
  else if(method=="medianfold.feature"){
   # This method centers each distribution in the median of the fold changes.
   # Method following hpbenton idea.
   average<-apply(originalIntensities, 1, median) # Changed to median and to hpbenton
method (still original code)
   # Replacing ceros for 0.0001 to avoid NaN or Inf values.
   average[which(average==0)]<-0.0001
   foldpersample<-sapply(1:dim(originalIntensities)[2], function(index){
     return(originalIntensities[,index]/average[index])
   })
   mfnormalizedIntensities<-t(sapply(1:dim(originalIntensities)[1], function(index){
     return(originalIntensities[index,]/median(foldpersample))
   }))
   #    colnames(mfnormalizedIntensities)<-colnames(originalIntensities)
#    mfnormalizedIntensities<-data.frame(mfnormalizedIntensities)
#    output<-cbind(meta, mfnormalizedIntensities)
#    output<-t(apply(output, 1, unlist))
   normalizedIntensities<-mfnormalizedIntensities
   normalizedIntensities<-t(apply(normalizedIntensities, 1, unlist))
  }
  else if(method=="medianfold.limma"){
   # This method centers each distribution in the median of the fold changes.
   # Method using the normalizeMedianValues() function from the limma package.
   mfnormalizedIntensities<-normalizeMedianValues(originalIntensities)
   mfnormalizedIntensities<-data.frame(mfnormalizedIntensities)
#    output<-cbind(meta, mfnormalizedIntensities)
   normalizedIntensities<-mfnormalizedIntensities
  } else
   return(NULL)
  if(columns1==min(blank.columns)){  # Blank samples are immediately after meta data.
   output<-cbind(xsetTable[,1:max(blank.columns)], normalizedIntensities)
  } else
   output<-cbind(cbind(meta, normalizedIntensities), xsetTable[,blank.columns])
```

```
 return(output)
}
```

removeFalsePositives.R

```
# The following function removes all the rows (features) that contain a feature that was either
UP
# or DOWN expressed in the control comparison.
# 'controles' contains the columns of the fold change and p-value values of the sham control
comparison.
## USAGE: todosPlasmoyControls_Substracted<-
removeFalsePositives(todosPlasmoyControls, 25:26)

removeFalsePositives<-function(tabla, controles, UPandDOWN=TRUE, fc.cutoff=1.5,
pval.cutoff=0.05){
	# Separate fold change and pvalue
	fc.control<-seq(controles[1], controles[length(controles)], 2)
	pval.control<-seq(controles[1]+1, controles[length(controles)], 2)
	# A fold change of 0 means that the Control in the pairwise comparison was 0 while
in the sample
	# was some value different than 0. While NaN means that the feature was not
detected in any of
	# the two samples compared.
	if(UPandDOWN==TRUE){
		vector<-apply(tabla, 1, function(linea){
			flag<-TRUE		# the row is not eliminated
			# First, NaN?
			if(is.na(linea[fc.control]) || is.na(linea[pval.control]))
				return(flag)		# No need to filter.
			if(linea[fc.control]>=fc.cutoff || linea[fc.control]<=1/fc.cutoff)
				if(linea[pval.control]<=pval.cutoff)
					flag<-FALSE		# FALSE means it was
overexpressed and must be removed.
			return(flag)
		})
	}
	if(UPandDOWN=="UP"){
	# This is in case it is specified that only features in the right column (Control) must
be UP regulated
	# in order to be removed from the table.
		vector<-apply(tabla, 1, function(linea){
			flag<-TRUE		# the row is not eliminated
			# First, NaN?
			if(is.na(linea[fc.control]) || is.na(linea[fc.control]))
				return(flag)		# No need to filter.
			if(linea[fc.control]>=fc.cutoff)
				if(linea[pval.control]<=pval.cutoff)
					flag<-FALSE		# FALSE means it was
overexpressed and must be removed.
			return(flag)
		})
	}
	if(UPandDOWN=="DOWN"){
	# This is in case it is specified that only features in the left column (Sample) must be
UP regulated
	# in order to be removed from the table.
		vector<-apply(tabla, 1, function(linea){
			flag<-TRUE		# the row is not eliminated
			# First, NaN?
			if(is.na(linea[fc.control]) || is.na(linea[fc.control]))
				return(flag)		# No need to filter.
```

```
                                        if(linea[fc.control]<=1/fc.cutoff)
                                                if(linea[pval.control]<=pval.cutoff)
                                                        flag<-FALSE         # FALSE means it was
overexpressed and must be removed.
                                        return(flag)
                        })
                }
                tabla<-tabla[which(vector),]
}
```

fcpvalue.R

```
# This function calculates the Fold change and P-value for each pairwise comparison.
## USAGE: todos<-fcpvalue(todos, samples=c(11:12), controls=c(9:10), "Pf")
fcpvalue<-function(tabla, samples, controls, name){
        temp<-t(apply(tabla, 1, function(linea){
                fc<-
median(as.numeric(linea[samples]))/median(as.numeric(linea[controls]))
#               if(fc<1)
#                       fc<-(-1/fc)
                obj<-try(t.test(x=linea[samples], y=linea[controls], var.equal=FALSE),
silent=TRUE)
                if (is(obj, "try-error"))
                        pvalue<-NA
                else
                        pvalue<-obj$p.value
#               pvalue<-t.test(x=linea[samples], y=linea[controls],
var.equal=FALSE)$p.value
                temp<-c(fc, pvalue)
                names(temp)<-c(paste(name, "fc", sep="."), paste(name, "pval", sep="."))
                return(temp)
        }))
        tabla<-cbind(tabla, temp)
}
```

mzmedFilter.R

```
# The following function returns only the features that have a mzmed between the given range.
## UPPlasmodium_mzfilter<-mzmedFilter(UPPlasmodium)
mzmedFilter<-function(tabla, mzmed=1, minimum=500, maximum=1000){
#       mzmed<-which(colnames(tabla)=="mzmed")
        buenos<-which(tabla[,mzmed]>=minimum & tabla[,mzmed]<=maximum)
        return(tabla[buenos,])
}
```

rtFilter.R

```
# This function filters the features with a retention time (rt) outside the established minimum
and
# maximum values.
rtFilter<-function(tabla, rt=4, minimum=3*60, maximum=14*60){
        buenos<-which(tabla[,rt]>=minimum & tabla[,rt]<=maximum)
        return(tabla[buenos,])
}
```

UPorDOWNregulated.R

```
# Create a new table with features that appeared UP regulated in both Plasmodium species
when compared
```

```
# to their corresponding control.
## USAGE: UPPlasmodium<-UPorDOWNregulated(todosPlasmoyControls_Substracted,
21:24, regulation="UP", fc.cutoff=1.5, pval.cutoff=0.05)
UPorDOWNregulated<-function(tabla, samples, regulation="UP", fc.cutoff=1.5,
pval.cutoff=0.05, biolRep.cutoff=0.66){
	fc.sample<-seq(samples[1], samples[length(samples)], 2)
	pval.sample<-seq(samples[1]+1, samples[length(samples)], 2)
	vector<-apply(tabla, 1, function(linea){
		flag<-TRUE		# the row is not eliminated
		# First, it is expressed in the sample? (different of NaN)
		if(sum(!is.na(linea[fc.sample]))<biolRep.cutoff*length(fc.sample) ||
sum(!is.na(linea[pval.sample]))<biolRep.cutoff*length(pval.sample))
				# I need a certain proportion of the samples to be UP regulated,
otherwise they are not markers for the samples.
					flag<-FALSE		# The feature is not present, must
be filtered out (FALSE).
		else if(regulation=="UP"){

	if(sum(linea[fc.sample]>=fc.cutoff)<biolRep.cutoff*length(fc.sample) ||
sum(linea[pval.sample]<=pval.cutoff)<biolRep.cutoff*length(pval.sample))
					flag<-FALSE		# The feature is not present above
the necessary fold change and under the pvalue thresholds, must be filtered out (FALSE).
		}
		else if(regulation=="DOWN"){

	if(sum(linea[fc.sample]<1/fc.cutoff)<biolRep.cutoff*length(fc.sample) ||
sum(linea[pval.sample]<=pval.cutoff)<biolRep.cutoff*length(pval.sample))
					flag<-FALSE		# The feature is not present above
the necessary fold change and under the pvalue thresholds, must be filtered out (FALSE).
		}
		return(flag)
	})
	tabla<-tabla[which(vector),]
}
```

Isotope.R

```
# The following function retrieves the closest possible isotope to given m/z and rt values,
considering
# a threshold for the m/z and the rt values. 'right' if TRUE indicates that the isotope (or adduct)
will
# be search in the M+1 range, is FALSE in the M-1 range. If intensityFilter is TRUE, the
isotope (adduct)
# must have a median intensity value smaller than the median intesity value of the given m/z.
# When using right=FALSE, the returned ratio will be greater than 1, confirming that the
isotope was found with a lower m/z value.
# USAGE: Isotope(856.5959, 484.005, todos, intensCols=c(11:16))
#mz=856.5959; rt=484.005; tabla=todos; right=TRUE; mzthreashold=0.0150;
rtthreashold=3.000; mzcolumn=1; rtcolumn=4; intensityFilter=TRUE; intensCols=c(11:16);
half=FALSE
Isotope<-function(mz, rt, tabla, right=TRUE, mzthreashold=0.0150, rtthreashold=3.000,
mzcolumn=1, rtcolumn=4, intensityFilter=TRUE, intensCols=NULL, half=FALSE){
 tabla<-tabla[which(tabla[,rtcolumn]>=rt-rtthreashold & tabla[,rtcolumn]<=rt+rtthreashold),]
# Selects rows with close rt value
 if(dim(tabla)[1]<1)
  return(print("This feature does not exist!!"))
 # Once we know the feature with the given mz is in 'tabla' we calculate the median intensity
for
 # M (mz) before we replace it.
 intM<-median(as.numeric(as.character(masCercano(mz, tabla, mzcolumn)[intensCols])))
 if(half==FALSE){
```

```
   amount<-1
  } else
   amount<-0.5
  if(right){  # Adds 'amount' to m/z value
   right<-amount
  } else
   right<--amount
  tabla<-tabla[which(tabla[,mzcolumn]>=mz+right-mzthreashold &
tabla[,mzcolumn]<=mz+right+mzthreashold),] # Selects rows with close m/z value
  if(is.null(dim(tabla)))
   return(print("No features where found!!"))
  if(dim(tabla)[1]<1)
     return(print("No features where found!!"))
  if(intensityFilter){  # The isotope (adduct) must have a median intensity value smaller than
the median intesity value of the given m/z.
   vectorints<-rep(NA, dim(tabla)[1])
   for(i in 1:dim(tabla)[1]){  # In case more than one row exists in 'tabla' the opportunity is
given to check if one of them has an expected intensity.
     intM1<-median(as.numeric(as.character(tabla[i, intensCols])))
     vectorints[i]<-intM1/intM
   }
   if(right>0){
     lower.values<-vectorints<1  # TRUE means this rows had an median intensity lower than
the intensity of the given m/z value
   } else
     lower.values<-vectorints>1
   candidate<-which(abs(tabla[which(lower.values),mzcolumn]-
(mz+right))==min(abs(tabla[which(lower.values),mzcolumn]-(mz+right))))
   # candidate contains the number of the row with the closest m/z value to the expected
isotope (adduct).
   print(paste(" Intensities ratio of the new m/z value (", tabla[candidate, mzcolumn], ") over
the original feature (", mz, ") is: ", vectorints[candidate], sep=""))
  }
  else {  # The closes mz value to the M+-right will be taken
   candidate<-which(abs(tabla[,mzcolumn]-(mz+right))==min(abs(tabla[,mzcolumn]-
(mz+right))))
  }
  tabla<-tabla[candidate,]
  if(dim(tabla)[1]<1)
   return(print("No features with lower intensity where found!!"))
  return(tabla)
}
```

isotopeFilter.R

```
# The following function returns a table with the features that correspond to M and not to M+1
values.
# Given that a standard Lipid feature has isotopes/adducts with M+1 and sometimes M+2,
features without this signals are also filtered when subfeatures=TRUE
# USAGE: isotopeFilter(UPPv, evalue=21, todos, intensCols=c(11:16))
# tabla.UPDOWN=UPPv; evalue=21; tabla.originals=todos; intensCols=c(11:16);
right=TRUE; mzthreashold=0.0150; rtthreashold=3.000; mzcolumn=1; rtcolumn=4;
intensityFilter=TRUE; subfeatures=TRUE; half=FALSE
isotopeFilter<-function(tabla.UPDOWN, evalue=21, tabla.originals, subfeatures=TRUE,
intensCols=NULL, right=TRUE, mzthreashold=0.0150, rtthreashold=3.000, mzcolumn=1,
rtcolumn=4, intensityFilter=TRUE, half=FALSE){
  # The following conversion will ensure that the variable 'tabla' will always have dimension
even
  # in the case where only one line is taken.
  tabla.originals<-as.data.frame(tabla.originals)
  # Ordering tabla.UPDOWN based on evalue
```

```
 tabla.UPDOWN<-tabla.UPDOWN[order(tabla.UPDOWN[ ,evalue]), ]
 candidatos<-t(apply(tabla.UPDOWN, 1, function(linea){
  #    linea<-tabla.UPDOWN[1,]
  linea<-as.numeric(linea)
  toreturn<-rep(NA, 6)
  elisotopo<-Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=FALSE, mzthreashold=mzthreashold, rtthreashold=rtthreashold,
mzcolumn=mzcolumn, rtcolumn=rtcolumn, intensityFilter=intensityFilter, half=half)
  # If searching to the left retrieves a result then this feature will not taken into account.
  if(!is.character(elisotopo))  # In case a data.frame is retrieved, then this feature should not be
considered.
    return(toreturn)
  if(!subfeatures){
    toreturn[1]<-linea[mzcolumn]
    toreturn[2]<-median(as.numeric(as.character(masCercano(linea[mzcolumn],
tabla.originals, mzcolumn)[1,intensCols])))
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, mzthreashold=mzthreashold, rtthreashold=rtthreashold,
mzcolumn=mzcolumn, rtcolumn=rtcolumn, intensityFilter=intensityFilter, half=half))
    if(!is.character(elisotopo)){
      toreturn[3]<-elisotopo[mzcolumn]
      toreturn[4]<-median(elisotopo[intensCols])
      elisotopo<-as.numeric(Isotope(elisotopo[mzcolumn], elisotopo[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, mzthreashold=mzthreashold, rtthreashold=rtthreashold,
mzcolumn=mzcolumn, rtcolumn=rtcolumn, intensityFilter=intensityFilter, half=half))
      if(length(elisotopo)>1){
        toreturn[5]<-elisotopo[mzcolumn]
        toreturn[6]<-median(elisotopo[intensCols])
      }
    }
  } else {  # subfeatures==TRUE, then it is requiered to have at least the M+1 feature
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, mzthreashold=mzthreashold, rtthreashold=rtthreashold,
mzcolumn=mzcolumn, rtcolumn=rtcolumn, intensityFilter=intensityFilter, half=half))
    if(length(elisotopo)>1){
      toreturn[1]<-linea[mzcolumn]
      toreturn[2]<-median(as.numeric(as.character(masCercano(linea[mzcolumn],
tabla.originals, mzcolumn)[1,intensCols])))
      toreturn[3]<-elisotopo[mzcolumn]
      toreturn[4]<-median(elisotopo[intensCols])
      elisotopo<-as.numeric(Isotope(elisotopo[mzcolumn], elisotopo[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, mzthreashold=mzthreashold, rtthreashold=rtthreashold,
mzcolumn=mzcolumn, rtcolumn=rtcolumn, intensityFilter=intensityFilter, half=half))
      if(length(elisotopo)>1){
        toreturn[5]<-elisotopo[mzcolumn]
        toreturn[6]<-median(elisotopo[intensCols])
      }
    }
  }
  # Returning: c(M_mz, M_medintensity, M1_mz, M1_medintensity, M2_mz,
M2_medintensity)
  return(toreturn)
 }))
 colnames(candidatos)<-c("M.mz", "M.int", "M1.mz", "M1.int", "M2.mz", "M2.int")
 return(candidatos)
}
```

ObtainingCandidates.R

```
# USAGE:
ObtainingCandidates(data.dir="C:/Users/a0119897/Desktop/pvivax/data/mzData_RBC/positiv
```

```
e_mode",
results.dir="C:/Users/a0119897/Desktop/pvivax/Tests/ObtainingCandidates_2ndPos",
normalization.method="medianfold.limma", blank.columns=c(9:10))
ObtainingCandidates<-function(data.dir, results.dir, normalization.method=FALSE,
blank.columns=NULL){
  setwd(results.dir)
  sink("Results.txt")
  print(Sys.time())
  library(xcms)
  library(limma)
  rawDataFiles <- list.files(data.dir, recursive = TRUE, full.names = TRUE)
  # Feature detection:
  xset <- xcmsSet(rawDataFiles, nSlaves=4, method="centWave", ppm=30,
peakwidth=c(15,30), snthresh=10,
                prefilter=c(2,1000), integrate=1, mzdiff=0.01, fitgauss=FALSE,
polarity="negative")
  # Retention time correction
  xset1<-retcor(xset, method="obiwarp", plottype=c("deviation"), profStep=1)
  # Alignment
  xset2<-group(xset1, bw=5,mzwid=0.015,minfrac=0.5)
  xset3<-fillPeaks(xset2)
  todos<-peakTable(xset3)
  print(" Printing table 'todos.tsv'")
  write.table(todos, "todos.tsv", quote=FALSE)
  if(normalization.method!=FALSE){
    todos.normalized<-featureNormalization(xset3, method=normalization.method,
blank.columns=c(9:10))
  }
  else
    todos.normalized<-todos
  print(" Printing table 'todos_normalized.tsv'")
  write.table(todos.normalized, "todos_normalized.tsv", quote=FALSE)

  errores<-c(errorPerc(todos, 9,10), errorPerc(todos, 11,12), errorPerc(todos, 13,14),
errorPerc(todos, 15,16),
           errorPerc(todos, 17,18), errorPerc(todos, 17,19), errorPerc(todos, 18,19))

  print(summary(errores))
  names(errores)<-c("Blank", "Pv1", "PV3", "PV5", "Retics1", "Retics2", "Retics3")
  print(errores)
  postscript("Reproducibility_feature_detection.eps", horizontal=FALSE, onefile=FALSE,
paper="special", width=10, height=8)
  barplot(errores, col=colorRampPalette(c("deepskyblue2", "limegreen"))(7),
main="Reproducibility in feature detection",
          xlab=NA, ylab="Reproducibility", las=2, ylim=c(0,100))
  dev.off()

  print(" Dimension of 'todos' table:")
  print(dim(todos))

  todosPv<-filterTechReps(todos, 11:12)
  todosPv<-filterTechReps(todosPv, 13:14)
  todosPv<-filterTechReps(todosPv, 15:16)
  print(" Dimension of the table after filtering technical replicates in 'Sample': ")
  print(dim(todosPv))
  todosPvRetics<-filterTechReps(todosPv, 17:19)
  print(" Dimension of the table after filtering technical replicates in 'Control': ")
  print(dim(todosPvRetics))
  todosPvReticsBlank<-filterTechReps(todosPvRetics, 9:10)
  print(" Dimension of the table after filtering technical replicates in 'Blank': ")
  print(dim(todosPvReticsBlank))
  blankTable<-fcpvalue(todosPvReticsBlank, samples=c(9:10), controls=c(11:19), "Blank")
```

```
  blankTable_Substracted<-removeFalsePositives(blankTable, 20:21, UPandDOWN="UP")
  print("Dimension of the table after removing False Positive features: ")
  print(dim(blankTable_Substracted))

  # Lines from 'todos.normalized' are taken if they were not filtered out by
removeFalsePositives()
  normalizedTable<-todos.normalized[which(rownames(todos.normalized) %in%
rownames(blankTable_Substracted)), ]
  normalizedTable<-fcpvalue(normalizedTable, samples=c(11:16), controls=c(17:19), "Pv")
  print(" Printing table 'normalizedTable_Subtracted.tsv'")
  write.table(normalizedTable, quote=FALSE, "normalizedTable_Subtracted.tsv")
  normalizedTable<-mzmedFilter(normalizedTable, minimum=300,
maximum=max(normalizedTable[,1]))
  print(" Dimension of the table after filtering for m/z value: ")
  print(dim(normalizedTable))
  normalizedTable<-rtFilter(normalizedTable)
  print(" Dimension of the table after filtering for retention time value: ")
  print(dim(normalizedTable))

  UPPv<-UPorDOWNregulated(normalizedTable, 20:21, regulation="UP", fc.cutoff=1.5,
pval.cutoff=0.05)
  UPRetic<-UPorDOWNregulated(normalizedTable, 20:21, regulation="DOWN",
fc.cutoff=1.5, pval.cutoff=0.05)
  print(" Dimension of the table containing UP regulated features in Pv: ")
  print(dim(UPPv))
  print(" Dimension of the table containing UP regulated features in Reticulocytes: ")
  print(dim(UPRetic))

  print(" Filtering mono-Isotopes in UPPv ")
  candidatos<-isotopeFilter(UPPv, evalue=21, todos, intensCols=c(11:16))
  tabla.UPDOWN<-UPPv[order(UPPv[ ,21]), ]
  tabla.UPPv<-tabla.UPDOWN[which(!is.na(candidatos[,1])), ]
  print(" Dimension of the table containing UP regulated  Mono-isotopes in Pv: ")
  print(dim(tabla.UPPv))
  print(" Printing table 'tabla.UPPv.tsv'")
  write.table(tabla.UPPv, quote=FALSE, "tabla.UPPv.tsv")

  print(" Plotting isotopic distributions in UPPv (Candidates)")
  candidatos.filtered<-candidatos[which(!is.na(candidatos[,1])), ]
  postscript("IsotopicDistributions_UPPv_Candidates.eps", horizontal=FALSE,
onefile=FALSE, paper="special", width=10, height=8)
  par(mfrow=c(3,5))
  for(i in 1:min(15, dim(tabla.UPPv)[1])){
    plot(NA, xlim=c(min(candidatos.filtered[i,c(1,3,5)], na.rm=TRUE)-1.5,
max(candidatos.filtered[i,c(1,3,5)], na.rm=TRUE)+1.5), ylim=c(0,
max(candidatos.filtered[i,c(2,4,6)], na.rm=TRUE)),
        xlab="m/z", ylab="Intensity", main=paste("Isotopic distribution
Retention time: ", masCercano(candidatos.filtered[i,1], UPPv, 1)[4], sep=""))
    lines(candidatos.filtered[i, c(1,3,5)], candidatos.filtered[i, c(2,4,6)], type="h", col=c("red",
"black", "black"))
  }
  dev.off()

  print(" Plotting isotopic distributions in UPPv (Eliminated)")
  eliminatedcandidates.index<-which(is.na(candidatos[,1]))
  ordenados<-UPPv[order(UPPv[,21]),]
  eliminatedcandidates.row<-ordenados[eliminatedcandidates.index,]

  tabla.UPDOWN=eliminatedcandidates.row; evalue=21; tabla.originals=todos;
intensCols=c(11:16); right=TRUE; mzthreashold=0.0150; rtthreashold=2.500; mzcolumn=1;
rtcolumn=4; intensityFilter=FALSE; subfeatures=FALSE;
  candidatos.elim<-t(apply(eliminatedcandidates.row, 1, function(linea){
```

```
   #    linea<-eliminatedcandidates.row[1,]
   linea<-as.numeric(linea)
   toreturn<-rep(NA, 8)  # space for a feature on the left.
   elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=FALSE, intensityFilter=FALSE))
   toreturn[1]<-elisotopo[mzcolumn]
   toreturn[2]<-median(elisotopo[intensCols])
   toreturn[3]<-linea[mzcolumn]
   toreturn[4]<-median(as.numeric(as.character(masCercano(linea[mzcolumn], tabla.originals,
mzcolumn)[1,intensCols])))
   elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE))
   toreturn[5]<-elisotopo[mzcolumn]
   toreturn[6]<-median(elisotopo[intensCols])
   elisotopo<-as.numeric(Isotope(elisotopo[mzcolumn], elisotopo[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE))
   toreturn[7]<-elisotopo[mzcolumn]
   toreturn[8]<-median(elisotopo[intensCols])
   # Returning: c(M_mz, M_medintensity, M1_mz, M1_medintensity, M2_mz,
M2_medintensity)
   return(toreturn)
 }))
 colnames(candidatos.elim)<-c("M_1.mz", "M_1.int", "M.mz", "M.int", "M1.mz", "M1.int",
"M2.mz", "M2.int")
 postscript("IsotopicDistributions_UPPv_Eliminated1.eps", horizontal=FALSE,
onefile=FALSE, paper="special", width=10, height=8)
 par(mfrow=c(3,5))
 for(i in 1:min(15, dim(candidatos.elim)[1])){
   plot(NA, xlim=c(min(candidatos.elim[i,c(1,3,5,7)], na.rm=TRUE)-1.5,
max(candidatos.elim[i,c(1,3,5,7)], na.rm=TRUE)+1.5), ylim=c(0,
max(candidatos.elim[i,c(2,4,6,8)], na.rm=TRUE)),
       xlab="m/z", ylab="Intensity", main=paste("Isotopic distribution
Retention time: ", masCercano(candidatos.elim[i,3], UPPv, 1)[4], sep=""))
   lines(candidatos.elim[i, c(1,3,5,7)], candidatos.elim[i, c(2,4,6,8)], type="h", col=c("black",
"red", "black", "black"))
 }
 dev.off()

 print(" Filtering mono-Isotopes in UPRetic ")
 candidatos<-isotopeFilter(UPRetic, evalue=21, todos, intensCols=c(17:19))
 tabla.UPDOWN<-UPRetic[order(UPRetic[ ,21]), ]
 tabla.UPRetic<-tabla.UPDOWN[which(!is.na(candidatos[,1])), ]
 print(" Dimension of the table containing UP regulated  Mono-isotopes in Reticulocytes: ")
 print(dim(tabla.UPRetic))
 print(" Printing table 'tabla.UPRetic.tsv'")
 write.table(tabla.UPRetic, quote=FALSE, "tabla.UPRetic.tsv")

 print(" Plotting isotopic distributions in UPRetic (Candidates)")
 candidatos.filtered<-candidatos[which(!is.na(candidatos[,1])), ]
 postscript("IsotopicDistributions_UPRetic_Candidates.eps", horizontal=FALSE,
onefile=FALSE, paper="special", width=10, height=8)
 par(mfrow=c(3,5))
 for(i in 1:min(15, dim(tabla.UPRetic)[1])){
   plot(NA, xlim=c(min(candidatos.filtered[i,c(1,3,5)], na.rm=TRUE)-1.5,
max(candidatos.filtered[i,c(1,3,5)], na.rm=TRUE)+1.5), ylim=c(0,
max(candidatos.filtered[i,c(2,4,6)], na.rm=TRUE)),
       xlab="m/z", ylab="Intensity", main=paste("Isotopic distribution
Retention time: ", masCercano(candidatos.filtered[i,1], UPRetic, 1)[4], sep=""))
   lines(candidatos.filtered[i, c(1,3,5)], candidatos.filtered[i, c(2,4,6)], type="h", col=c("red",
"black", "black"))
 }
 dev.off()
```

```
  print(" Identifying which features inside 'tabla.UPPv' that have 'half-features' only at the
right")
  candidatos.half<-isotopeFilter(tabla.UPPv, evalue=21, todos, intensCols=c(11:16),
half=TRUE)
  print(" Printing the candidates with a feature in the M+0.5 range: ")
  print(candidatos.half[!is.na(candidatos.half[,1]),])
  print(" Identifying which features inside 'tabla.UPPv' that have 'half-features' at the left")
  eliminatedcandidates.index<-which(is.na(candidatos.half[,1]))
  ordenados<-tabla.UPPv[order(tabla.UPPv[,21]),]
  eliminatedcandidates.row<-ordenados[eliminatedcandidates.index,]

  tabla.UPDOWN=eliminatedcandidates.row; evalue=21; tabla.originals=todos;
intensCols=c(11:16); right=TRUE; mzthreashold=0.0150; rtthreashold=2.500; mzcolumn=1;
rtcolumn=4; intensityFilter=FALSE; subfeatures=FALSE; half=TRUE
  candidatos.elim<-t(apply(eliminatedcandidates.row, 1, function(linea){
    #    linea<-eliminatedcandidates.row[1,]
    linea<-as.numeric(linea)
    toreturn<-rep(NA, 8)  # space for a feature on the left.
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=FALSE, intensityFilter=FALSE, half=half))
    toreturn[1]<-elisotopo[mzcolumn]
    toreturn[2]<-median(elisotopo[intensCols])
    toreturn[3]<-linea[mzcolumn]
    toreturn[4]<-median(as.numeric(as.character(masCercano(linea[mzcolumn], tabla.originals,
mzcolumn)[1,intensCols])))
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE, half=half))
    toreturn[5]<-elisotopo[mzcolumn]
    toreturn[6]<-median(elisotopo[intensCols])
    elisotopo<-as.numeric(Isotope(elisotopo[mzcolumn], elisotopo[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE, half=half))
    toreturn[7]<-elisotopo[mzcolumn]
    toreturn[8]<-median(elisotopo[intensCols])
    # Returning: c(M_mz, M_medintensity, M1_mz, M1_medintensity, M2_mz,
M2_medintensity)
    return(toreturn)
  }))
  colnames(candidatos.elim)<-c("M_1.mz", "M_1.int", "M.mz", "M.int", "M1.mz", "M1.int",
"M2.mz", "M2.int")
  print(" Printing the candidates with a feature in the M-0.5 range: ")
  print(candidatos.elim[!is.na(candidatos.elim[,1]),])

  print(" Identifying which features inside 'tabla.UPRetic' that have 'half-features' only at the
right")
  candidatos.half<-isotopeFilter(tabla.UPRetic, evalue=21, todos, intensCols=c(11:16),
half=TRUE)
  print(" Printing the candidates with a feature in the M+0.5 range: ")
  print(candidatos.half[!is.na(candidatos.half[,1]),])
  print(" Identifying which features inside 'tabla.UPRetic' that have 'half-features' at the left")
  eliminatedcandidates.index<-which(is.na(candidatos.half[,1]))
  ordenados<-tabla.UPRetic[order(tabla.UPRetic[,21]),]
  eliminatedcandidates.row<-ordenados[eliminatedcandidates.index,]

  tabla.UPDOWN=eliminatedcandidates.row; evalue=21; tabla.originals=todos;
intensCols=c(11:16); right=TRUE; mzthreashold=0.0150; rtthreashold=2.500; mzcolumn=1;
rtcolumn=4; intensityFilter=FALSE; subfeatures=FALSE; half=TRUE
  candidatos.elim<-t(apply(eliminatedcandidates.row, 1, function(linea){
    #    linea<-eliminatedcandidates.row[1,]
    linea<-as.numeric(linea)
    toreturn<-rep(NA, 8)  # space for a feature on the left.
```

```
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=FALSE, intensityFilter=FALSE, half=half))
    toreturn[1]<-elisotopo[mzcolumn]
    toreturn[2]<-median(elisotopo[intensCols])
    toreturn[3]<-linea[mzcolumn]
    toreturn[4]<-median(as.numeric(as.character(masCercano(linea[mzcolumn], tabla.originals,
mzcolumn)[1,intensCols])))
    elisotopo<-as.numeric(Isotope(linea[mzcolumn], linea[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE, half=half))
    toreturn[5]<-elisotopo[mzcolumn]
    toreturn[6]<-median(elisotopo[intensCols])
    elisotopo<-as.numeric(Isotope(elisotopo[mzcolumn], elisotopo[rtcolumn], tabla.originals,
intensCols=intensCols, right=TRUE, intensityFilter=FALSE, half=half))
    toreturn[7]<-elisotopo[mzcolumn]
    toreturn[8]<-median(elisotopo[intensCols])
    # Returning: c(M_mz, M_medintensity, M1_mz, M1_medintensity, M2_mz,
M2_medintensity)
    return(toreturn)
  }))
  colnames(candidatos.elim)<-c("M_1.mz", "M_1.int", "M.mz", "M.int", "M1.mz", "M1.int",
"M2.mz", "M2.int")
  print(" Printing the candidates with a feature in the M-0.5 range: ")
  print(candidatos.elim[!is.na(candidatos.elim[,1]),])


  print(" DONE!! ")
  print(Sys.time())
  sink()
}
```

extractingSpectra.R

```
# Function to extract the spectra corresponding to a given m/z and rt, given respective
thresholds.
# The retention time is always considered in seconds.
## USAGE:
extractingSpectra(mgf.file="MSMS/mspepsearchC8/C8_PV_targetedMSMSpos_6and4_3collr
1.mgf", candidates.file="Candidates_PositiveMode.UPPv_wSample.tsv", mz.col=1, rt.col=2,
output.file="MSMS/mspepsearchC8/December/C8_PV_targetedMSMSpos_Filtered.mgf")
extractingSpectra<-function(mgf.file, candidates.file, mz.col=1, rt.col=2, output.file,
mz.threshold=0.015, rt.threshold=30){
  # mgf.file="MSMS/mspepsearchC8/C8_PV_targetedMSMSpos_6and4_3collr1.mgf";
candidates.file="Candidates_PositiveMode.UPPv_wSample.tsv"; mz.col=1; rt.col=2;
output.file="MSMS/mspepsearchC8/December/C8_PV_targetedMSMSpos_Filtered.mgf";
mz.threshold=0.015; rt.threshold=30
  # Reading files:
  mgf.file<-readLines(mgf.file)
  candidates.file<-read.table(candidates.file, header=TRUE)
  # Indexing the mgf file:
  PEPMASS.lines<-grep("PEPMASS=", mgf.file)
  RTINSECONDS.lines<-grep("RTINSECONDS=", mgf.file)
  PEPMASS.values<-as.numeric(sub("PEPMASS=(.+) ", "\\1", mgf.file[PEPMASS.lines],
perl=TRUE))
  RTINSECONDS.values<-as.numeric(sub("RTINSECONDS=(.+)", "\\1",
mgf.file[RTINSECONDS.lines], perl=TRUE))
  # Finding MS/MS spectra for the given candidates:
  subset<-NULL
  for(i in 1:dim(candidates.file)[1]){
    goodmz<-which(abs(PEPMASS.values-candidates.file[i,mz.col])<=mz.threshold)
    goodmzrt<-which(abs(RTINSECONDS.values[goodmz]-
candidates.file[i,rt.col])<=rt.threshold)
```

```
   goodmzrt.rtlines<-RTINSECONDS.lines[goodmz][goodmzrt]
   for(j in 1:length(goodmzrt.rtlines)){
    begin.ions<-goodmzrt.rtlines[j]-4
    for (end.ions in (goodmzrt.rtlines[j]+2):length(mgf.file))
     if(mgf.file[end.ions]=="END IONS")
      break
    subset<-c(subset, mgf.file[begin.ions:end.ions])
    #writeLines(subset, out.mgf)
   }
  }
  titulos<-grep("TITLE=", subset)
  if(length(table(table(titulos)))>1)
   print(" WARNING! At least one spectra is repeated due to overlaping in the Candidates file
or by an unknown reason.")
  writeLines(subset, output.file)
}
```

getRTmins.R

```
# Function to extract the Retention Time in seconds from the Unknown column
## USAGE: Res.Pv.pos.lb<-getRTmins(Res.Pv.pos.lb)
getRTmins<-function(tsv){
 #  tsv<-Res.Pv.pos.lb
 rtstring<-sapply(tsv$Unknown, function(unknown){
  rtstring<-strsplit(as.character(unknown), " at ")[[1]][2]
  rtstring<-as.numeric(strsplit(rtstring, " mins ")[[1]][1])
  return(rtstring)
 })
 rtstring<-data.frame(rt.min=rtstring)
 tsv<-cbind(tsv, rtstring)
 return(tsv)
}
```

concatenate.tsvs.R

```
# Function to concatenate two .tsv files using the same query .mgf but different databases in
MSPepSearch
## USAGE: Res.Pv.pos<-concatenate.tsvs(Res.Pv.pos.lb, Res.Pv.pos.custom)
concatenate.tsvs<-function(tsv1, tsv2){
#  tsv1<-Res.Pv.pos.lb; tsv2<-Res.Pv.pos.custom
 tsv<-rbind(tsv1, tsv2)
 tsv<-tsv[order(tsv$rt.min),]
#  tsv<-temp2
 unique.queries<-unique(tsv$Num)
 tsv.ordenado<-tsv[0,]
 for(i in unique.queries){
  subset<-tsv[which(tsv$Num==i),]
  tsv.ordenado<-rbind(tsv.ordenado, subset[order(-subset$Rev.Dot),])
 }
 return(tsv.ordenado)
}
```

LipidClassAssignment.R

```
# Function to identify the safest information from each precursor.
# The degree of confidence can be:
# 1. Lipid class (MGDG, PA, PC, DG, PC, PE, PS, CE, Plasmenyl-PC, and probably others).
# 2. Subclass (GPser, GPCho, GPA, GPEtn, Cholesteryl ester).
# 3. Total carbons in the fatty acyls and number of double bonds (PC 36:5, MGDG 30:3,
plasmenyl-PC 36:4,
#     PA 41:0, PE 37:4, CE(16:2), PS 41:3).
```

```
# 4. Distribution of the carbons and double bonds in the fatty acyls (GPEtn(17:2/20:2), PC(P-
20:0/20:4),
#     GPCho(20:4/20:4), MGDG(15:0/17:1), DG(12:0/20:0/0:0), GPA(17:0/24:0),
GPSer(15:1/26:2), PE(P-20:0/15:0)).
## USAGE: Ident.Retic.pos<-LipidClassAssignment(Res.Retic.pos)
LipidClassAssignment<-function(tsv){
# tsv<-Res.Retic.pos
# head(tsv)
  tsv<-tsv[which(tsv$Rev.Dot>=300),]
  unique.queries<-unique(tsv$Num)
#  Assigned<-data.frame(Num=NA, Precursor.m.z=NA, Library=NA, Rev.Dot=NA, LC=NA,
Adduct=NA, sns=NA, sn1=NA, sn2=NA, sn3=NA, RTsec=NA)
  Assigned<-rep(NA, 11)
  for(i in unique.queries){
    subset<-tsv[which(tsv$Num==i),]
    # If only one of the hits in 'subset' has a 'Rev.Dot' value higher or equal to 600, the
assignment
    # given to this hit is copied.
    if(length(sub.subset<-which(subset$Rev.Dot>=600))==1){
      linea<-subset[sub.subset,c(1,3,5,13,14,22)]
      lipid<-splitPeptide(linea[1,5])
      Assigned<-rbind(Assigned, cbind(linea[1,c(1:4,6)], t(lipid)))
    }
    # If more than one of the hits in 'subset' has a 'Rev.Dot' value higher or equal to 600, the
shared
    # assignment is copied.
    else if(length(sub.subset) > 1){
      linea<-subset[which.max(subset$Rev.Dot[sub.subset]),c(1,3,5,13,14,22)]
      df<-t(apply(subset, 1, function(linea){
        splitPeptide(linea[14])
      }))
      lipid<-rep(NA, 7)
      lipid[2]<-df[which.max(subset$Rev.Dot[sub.subset]),2]
      if(length(unique(df[,1]))==1){
        lipid[1]<-df[1,1]
        # The 'subclass' is checked before the number of carbons in the fatty acyls
        if(length(unique(df[,4]))==1){
          lipid[4]<-df[1,4]
          if(length(unique(df[,3]))==1){
            lipid[3]<-df[1,3]
            if(length(unique(df[,5]))==1){
              lipid[5]<-df[1,5]
              if(length(unique(df[,6]))==1){
                lipid[6]<-df[1,6]
                if(length(unique(df[,7]))==1){
                  lipid[7]<-df[1,7]
                }
              }
            }
          }
        }
      }
      Assigned<-rbind(Assigned, cbind(linea[1,c(1:4,6)], t(lipid)))
    }
    # If none of the hits in 'subset' has a 'Rev.Dot' value higher or equal to 600, nothing is
copied.
    else {
      linea<-subset[which.max(subset$Rev.Dot),c(1,3,5,13,22)]
      Assigned<-rbind(Assigned, cbind(linea, t(rep(NA, 7))))
    }
  }
  Assigned<-Assigned[2:dim(Assigned)[1],]
```

```
  colnames(Assigned)<-c(colnames(Assigned)[1:5], "LC", "Adduct", "sns", "subclass", "sn1",
"sn2", "sn3")
  return(Assigned)
}
```

clusteringSpectra.R

```
# Function to make clusters of mass spectra given a table coming from
LipidClassAssignment() and a mz
# threshold value.
## USAGE: Ident.Retic.clustered<-clusteringSpectra(tabla=Ident.Retic, mz.threshold=0.02)
clusteringSpectra<-function(tabla, mz.threshold=0.015){
#  tabla=Ident.Retic; mz.threshold=0.02;
  tabla<-tabla[order(tabla$Precursor.m.z),]
  tabla$diff.adducts<-rep(1, dim(tabla)[1])
  for(i in 1:(dim(tabla)[1]-1)){
#   i<-2
    # Creating a subtable that contains precursors very close to the row 'i'
    tabla.remaining<-tabla[(i+1):dim(tabla)[1],]
    to.check<-which((tabla.remaining$Precursor.m.z-tabla[i,2])<=mz.threshold)+i
    if(length(to.check)>0){
      for(j in to.check){
# j<-to.check[1]
       # Verifying if this row belongs to a previous identified cluster
       if(tabla$diff.adducts[j]>0){
         # Coming from a different adduct? If it comes from the same adduct it is not important,
they will
         # only have slightly different retention times (probably)
         if(tabla[i,7]!=tabla[j,7]){
           # Having the same lipid assignation?
           if(setequal(tabla[i,c(8:12)], tabla[j, c(8:12)])){
             tabla$diff.adducts[i]<-tabla$diff.adducts[i]+1
           }
         } else {
           # Hits with the same adduct, but with different assignation? (coming from different RT)
           if(setequal(tabla[i,c(8:12)], tabla[j, c(8:12)])){
             tabla$diff.adducts[j]<-0
           }
         }
       }
      }
    }
  }
  return(tabla)
}
```

splitPeptide.R

```
# Function to split the string of the 'Peptide' name into the vector of 6 values
## USAGE: splited<-splitPeptide(elString)
splitPeptide<-function(elString){
#  elString<-subset[1,14]
  elString<-as.character(elString)
  temp<-strsplit(elString, " ")[[1]]
  Peptide.LC<-temp[1]
  Peptide.Adduct<-strsplit(temp[3], ";")[[1]][1]
  Peptide.sns<-strsplit(temp[2], ";")[[1]][1]
  Peptide.subclass<-sub("(\\w+)\\(.+", "\\1", temp[4], perl=TRUE)
  temp<-sub("(.+)\\(.+\\)(.+)", "\\1\\2", temp[4])
  temp<-sub("(.+)\\(.+\\)(.+)", "\\1\\2", temp)
```

```
 temp<-sub("(.+)\\(.+\\)(.+)", "\\1\\2", temp)
 temp<-sub(".+\\((.+)\\)", "\\1", temp)
 temp<-strsplit(temp, "/")[[1]]
 Peptide.sn1<-temp[1]
 Peptide.sn2<-temp[2]
 if(Peptide.LC=="TG" || Peptide.LC=="DG" || Peptide.LC=="MG"){
  Peptide.sn3<-temp[3]
 } else
  Peptide.sn3<-NA
 elString<-
c(Peptide.LC,Peptide.Adduct,Peptide.sns,Peptide.subclass,Peptide.sn1,Peptide.sn2,Peptide.sn3
)
 return(elString)
}
```

getMGF.R

```
# Function to generate a mgf file with the spectra from a given mz/rt values
## USAGE: getMGF(mz="836.5566406", rt="6.92803333333333",
in.mgf="C8_PV_targetedMSMSneg_6and4_3collr1.mgf", out.mgf="salida.mgf")
getMGF<-function(mz, rt, in.mgf, out.mgf="archivo_salida.mgf"){
#  mz=836.5566406; rt="6.92803333333333";
in.mgf="Retic_targetedMSMSpos_6and4_3coll.mgf"; out.mgf<-"archivo_salida.mgf"
 in.mgf<-readLines(in.mgf)
 elString<-paste("MS/MS of ", mz, " 1+ at ", rt, sep="")
 TITLE<-grep(elString, in.mgf, fixed=TRUE)
 if(length(TITLE)>1)
  print(" WARNING: Something weird happened!!")
 begin.ions<-TITLE-3
 for (end.ions in (TITLE+3):length(in.mgf))
  if(in.mgf[end.ions]=="END IONS")
   break
 subset<-in.mgf[begin.ions:end.ions]
 writeLines(subset, out.mgf)
}
```

getMGF.feelinglucky.R

```
# Function to generate a mgf file with the spectra from a given mz/rt approximate values
## USAGE: getMGF.feelinglucky(mz="836.5566406", rt="6.92803333333333",
in.mgf="Retic_targetedMSMSpos_6and4_3coll.mgf", out.mgf="salida.mgf")
getMGF.feelinglucky<-function(mz, rt, in.mgf, out.mgf="archivo_salida.mgf"){
 #  mz=836.5566; rt=6.92803; in.mgf="Retic_targetedMSMSpos_6and4_3coll.mgf";
out.mgf<-"archivo_salida.mgf"
 in.mgf<-readLines(in.mgf)
 TITLE<-grep(paste(".+MS/MS of ", mz, ".+ 1\\+ at ", rt, sep=""), in.mgf, perl=TRUE)
 if(!any(length(TITLE))){
  recorta<-function(number){
   tam<-nchar(number)
   return(substr(number, 1, tam-1))
  }
  mz2<-recorta(mz)
  TITLE<-grep(paste(".+MS/MS of ", mz2, ".+ 1\\+ at ", rt, sep=""), in.mgf, perl=TRUE)
  if(!any(length(TITLE))){
   rt2<-recorta(rt)
   TITLE<-grep(paste(".+MS/MS of ", mz, ".+ 1\\+ at ", rt2, sep=""), in.mgf, perl=TRUE)
   if(!any(length(TITLE))){
    TITLE<-grep(paste(".+MS/MS of ", mz2, ".+ 1\\+ at ", rt2, sep=""), in.mgf, perl=TRUE)
    if(!any(length(TITLE))){
     print(" mz and rt values do not match with any spectra!")
```

```
        return(NULL)
       }
      }
     }
    }
    if(length(TITLE)>1)
     print(" WARNING: More than one hit matched!")
    begin.ions<-TITLE-3
    for (end.ions in (TITLE+3):length(in.mgf))
     if(in.mgf[end.ions]=="END IONS")
      break
    subset<-in.mgf[begin.ions:end.ions]
    writeLines(subset, out.mgf)
}
```

generateMGFs.R

```
# Function to generate a .mgf file for each of the rows of a 'Ident.*.clustered' table.
# This requires a directory with the name spectra in the current directory.
## USAGE: generateMGFs(tabla=Ident.Pv.clustered, sample="Pv",
pos_mgf="C8_PV_targetedMSMSpos_6and4_3collr1.mgf",
neg_mgf="C8_PV_targetedMSMSneg_6and4_3collr1.mgf")
generateMGFs<-function(tabla, sample, pos_mgf, neg_mgf){
 tabla<-tabla[order(-tabla$Rev.Dot),]
 for(i in 1:dim(tabla)[1]){
  if(any(grep("pos", tabla[i,3]))){
   in.mgf<-pos_mgf
  out.mgf<-"pos.mgf"
  } else {
   in.mgf<-neg_mgf
   out.mgf<-"neg.mgf"
  }
  out.mgf<-paste(sample, i, "_", tabla[i,9], tabla[i,1], out.mgf, sep="")
  getMGF.feelinglucky(mz=tabla[i,2], rt=tabla[i,5], in.mgf, paste("spectra", out.mgf, sep="/"))
 }
}
```

neutralLossScanner.R

```
# Function to identify product ions and neutral loss features from a multi-mgf file
# The argument 'mode' receives either "Neg" or "Pos" strings only.
## USAGE: neutralLossScanner<-
neutralLossScanner(MGF="C8_PV_targetedMSMSneg_6and4_3collr1.mgf",
report="PISandNLSReport_PvNeg.txt", mode="Neg")
neutralLossScanner<-function(MGF, mode,
DB="/Users/a0119897/Desktop/pvivax/data/NeutralLoss_values.txt", mz.threshold=0.015,
report="report.txt"){
 write.table(" This is the report for PIS and NLS", report, append=FALSE, col.names=FALSE,
quote=FALSE, row.names=FALSE)
 write.table("", report, append=TRUE, col.names=FALSE, quote=FALSE,
row.names=FALSE)
 DB<-read.table(DB, header=TRUE, sep="\t", comment.char = "#")
 forPIS<-DB[which(DB$Mode==mode & DB$msms.scantype=="PIS"),]
 forNLS<-DB[which(DB$Mode==mode & DB$msms.scantype=="NLS"),]
 MGF<-readLines(MGF)
 spectra.beginings<-grep("BEGIN IONS", MGF)
 spectra.endings<-grep("END IONS", MGF)
 spectra.num<-length(spectra.beginings)
 spectra<-vector("list", spectra.num)
 count<-1
```

```
 # Separating the spectra
 for(i in 1:spectra.num){
  spectra[[count]]<-MGF[spectra.beginings[i]:spectra.endings[i]]
  count<-count+1
 }
 # Identification by Product Ion Scan
 withHitsPIS<-unlist(lapply(spectra, function(spectra){
  # spectra<-spectra[[5061]]
  # Obtaining only the m/z values
  Ions<-spectra[6:(length(spectra)-1)]
  Ions<-sub("\\t.+", "", Ions)
  # Comparing with PIS values from DB
  positives<-sapply(forPIS$exact.mass, function(mz.db){
   return(abs(mz.db-as.numeric(Ions))<=mz.threshold)
  })
  if(sum(positives)>0){
   if(sum(rowSums(positives)>1)>0)
    print(paste(" Something weird happened in the spectra number ", i, sep=""))
   found<-as.character(forPIS$Lipid.class[which(colSums(positives)>0)])
#    forPIS[which(colSums(positives)>0),]
   return(c(spectra[2], spectra[4], found))
  } else{
   return(NULL)
  }
#return(sum(positives))
 }))
 # Identification by Neutral Loss Scan
 withHitsNLS<-unlist(lapply(spectra, function(spectra){
  # spectra<-spectra[[440]]
  precursor<-as.numeric(sub("PEPMASS=(\\d+\\.\\d+)\\s+", "\\1", spectra[2]))
  # Obtaining only the m/z values
  Ions<-spectra[6:(length(spectra)-1)]
  Ions<-sub("\\t.+", "", Ions)
  # Comparing with NLS values from DB
  positives<-sapply(forNLS$exact.mass, function(mz.db){
   return(abs(mz.db-(precursor-as.numeric(Ions)))<=mz.threshold)
  })
  if(sum(positives)>0){
   if(sum(rowSums(positives)>1)>0)
    print(paste(" Something weird happened in the spectra number ", i, sep=""))
   found<-as.character(forNLS$Lipid.class[which(colSums(positives)>0)])
#    forNLS[which(colSums(positives)>0),]
   return(c(spectra[2], spectra[4], found))
  } else{
   return(NULL)
  }
#return(sum(positives))
 }))

 # Identifying hits in PIS and NLS regardless if the identification of the Lipid Class was the
same
 HitsPIS<-grep("PEPMASS=", withHitsPIS)
 HitsInPISandNLS<-NULL
 for(i in HitsPIS)
 # i<-2
  if(any(grep(withHitsPIS[i+1], withHitsNLS)))
   HitsInPISandNLS<-c(HitsInPISandNLS, withHitsPIS[i+1])
 write.table(paste(length(HitsInPISandNLS), " where identified in PIS and NLS "), report,
append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
 write.table("", report, append=TRUE, col.names=FALSE, quote=FALSE,
row.names=FALSE)
```

```
 # Clustering the precursors to identify spectra with more than one hit.
 # Then checking if the assignments were consistent.
 AllHits<-c(withHitsPIS, withHitsNLS)
 AllHits.PEPMASS<-grep("PEPMASS=", AllHits)
 AllHits.num<-length(AllHits.PEPMASS)
 TheHits<-vector("list", AllHits.num)
 count<-1
 for(i in 1:AllHits.num){
  if(i!=AllHits.num){
   TheHits[[count]]<-AllHits[AllHits.PEPMASS[i]:(AllHits.PEPMASS[i+1]-1)]
  } else{
   TheHits[[count]]<-AllHits[AllHits.PEPMASS[i]:length(AllHits)]
  }
  count<-count+1
 }
 temp<-unlist(lapply(TheHits, function(x)@x[1]}))          # Substitute the @ for a {
 Precursors<-names(rev(sort(table(temp))))
 # Idea: por cada precursor unico ver que asignaciones se encontraron, si son congruentes se reporta, si no, se reporta como ambiguo.
 # The following takes into account the possible multiple hits for the same spectra (same precursor, same retention time)
 for(i in 1:length(Precursors)){
# i<-1
  spectras<-grep(Precursors[i], TheHits)
  subSpectras<-lapply(spectras, function(x){
   return(TheHits[[x]])
  })
  tabla.assign<-table(unlist(lapply(subSpectras, function(x){
   maximo<-length(x)
   return(x[3:maximo])
  })))
#  write.table(" This is the report for PIS and NLS", report, append=FALSE, col.names=FALSE, quote=FALSE, row.names=FALSE)
  write.table(Precursors[i], report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
  write.table(tabla.assign, report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
  write.table("", report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
 }

 IdentifiedTwice<-unlist(lapply(TheHits, function(spectra){
# spectra<-TheHits[[96]]
  if(length(spectra)==4){
   if(any(grep(spectra[3], spectra[4]))){
    return(c(spectra[1], spectra[2], spectra[3]))
   } else if(any(grep(spectra[4], spectra[3]))){
    return(c(spectra[1], spectra[2], spectra[4]))
   } else {
    write.table(paste(" Ambiguous identification or double lipid in ", spectra[2], sep=""), report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
    return(NULL)
   }
  } else if (length(spectra)==5){
   opciones<-c(spectra[3], spectra[4], spectra[5])
   if(table(nchar(opciones))>1 & sum(table(opciones)>1)>0){
    # At least there are two objects with the same length and are not the same Lipid Class
    write.table(paste(" Ambiguous identification or double lipid in ", spectra[2], sep=""), report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
    return(NULL)
   }
```

```
    primero<-opciones[which.min(nchar(opciones))]
    tercero<-opciones[which.max(nchar(opciones))]
    segundo<-opciones[c(-which.min(nchar(opciones)), -which.max(nchar(opciones)))]
    if(any(grep(primero, segundo)) | any(grep(primero, tercero))){
      return(c(spectra[1], spectra[2], primero))
    } else if(any(grep(segundo, tercero)) | any(grep(segundo, primero))){
      return(c(spectra[1], spectra[2], segundo))
    } else if(any(grep(tercero, primero)) | any(grep(tercero, segundo))){
        return(c(spectra[1], spectra[2], tercero))
    } else {
      write.table(paste(" Ambiguous identification or double lipid in ", spectra[2], sep=""),
report, append=TRUE, col.names=FALSE, quote=FALSE, row.names=FALSE)
      return(NULL)
    }
   }
  }))
  write.table("", report, append=TRUE, col.names=FALSE, quote=FALSE,
row.names=FALSE)
  write.table("Precursors with a second molecule indetification: ", report, append=TRUE,
col.names=FALSE, quote=FALSE, row.names=FALSE)
  write.table(IdentifiedTwice, report, append=TRUE, col.names=FALSE, quote=FALSE,
row.names=FALSE)
  print(" Done")
  return()
}
```

Appendix C

List of m/z values used to perform the Tandem MS experiment

Candidates from the *P. vivax*-infected samples in positive mode.

mz	Pv.fc	Pv.pval
856.595899	13.952269	3.49E-05
830.580989	3.0227331	0.000106
864.630715	Inf	0.00010929
1487.01798	Inf	0.00012066
766.584141	2.2597152	0.00017286
692.539355	15.398755	0.00025878
754.551404	12.230668	0.00029472
1590.12188	Inf	0.00034024
868.662778	5.0383312	0.00035809
1588.11015	Inf	0.0005086
1463.01838	Inf	0.00052355
1449.03543	Inf	0.0006573
750.542885	2.6335975	0.00069983
718.588676	14.891729	0.00072916
819.597013	3.2862761	0.00082192
1004.63396	1.6584495	0.00083519
1439.02002	Inf	0.00092784
1000.60298	2.0130696	0.00111595
1564.11426	22.058069	0.00136471
774.541916	Inf	0.00139384
720.603709	7.8380775	0.00141889
714.521235	4.488401	0.00142359
716.539165	13.9351	0.0014344
980.635546	4.7575506	0.00146309
1296.9799	Inf	0.00148169
575.522075	Inf	0.00158734
844.663867	84.930913	0.00166426
904.599879	3.2722348	0.00167646
863.56517	Inf	0.0017033
816.63299	2.3111484	0.00173442
817.580491	5.5994355	0.00190877
634.559006	898.32143	0.00195081
902.582811	Inf	0.00204318
1544.14057	53.855888	0.00229509
880.600461	Inf	0.0023182
1570.15721	20.46741	0.00232964
1528.13282	Inf	0.00233512
645.47478	2.0521741	0.00237814
1546.15872	Inf	0.00256802
808.598716	1.7693671	0.00266626
601.536123	5.7001576	0.00268396
706.606365	2.0919391	0.00272273
780.568213	3.8109109	0.00280168

mz	Pv.fc	Pv.pval
834.61273	3.5482265	0.00293001
1425.0329	6.5050709	0.00321665
703.59132	1.5658117	0.00321965
1540.1169	22.750913	0.00326524
806.58356	2.0796806	0.00358534
956.63535	5.7403729	0.0035974
744.57533	3.7660237	0.00368244
982.65146	2.5540982	0.00370433
1468.1172	103.26238	0.00382042
786.61558	4.0936344	0.00386658
788.54303	2.9674187	0.00393016
568.58458	2.5902995	0.00397072
701.57395	2.3049493	0.0040769
744.56482	3.2164505	0.00420329
878.58304	Inf	0.00434872
369.37402	2.2386226	0.00451886
788.62135	4.0029176	0.00463973
742.55295	3.2952675	0.00470451
599.52135	Inf	0.00478747
836.62653	2.2598244	0.00487221
599.52145	4.3046882	0.00487905
742.57959	2.7179709	0.00497773
1492.1167	Inf	0.00506632
660.57392	6.5912957	0.00510306
756.56877	3.7993304	0.00537963
648.64426	2.31537	0.00541326
1516.117	49.058067	0.00561275
865.57993	2.3863679	0.00579398
1494.1317	6.4421761	0.00598426
815.5678	2.4362743	0.00602275
762.61598	3.2220494	0.00655519
758.5849	3.0231728	0.00712553
740.53883	2.3449759	0.00743697
801.69639	1.6343348	0.00747159
748.63428	2.4634838	0.00784872
900.56827	Inf	0.00834924
788.63547	2.3637189	0.00841856
734.58531	2.2386208	0.00864158
1524.0832	Inf	0.00896531
772.59841	2.1547451	0.00941585
1520.1467	3.0661007	0.01051051
625.53568	2.1003204	0.01122213
1498.0684	Inf	0.01147119

mz	Pv.fc	Pv.pval
812.63117	2.420992	0.0119153
952.60576	1.719475	0.0120025
613.51464	1.593654	0.0126804
1522.0676	Inf	0.0127669
822.64702	1.79117	0.0137918
774.64905	1.826896	0.014053
796.60014	1.533133	0.0154807
340.37994	2.255228	0.0157024
794.61602	1.61953	0.0162163
662.58818	Inf	0.016569
718.55514	1.859272	0.0167094
841.58138	1.580126	0.0177887
636.57293	17.7375	0.0207109
839.56596	1.602346	0.0208957
764.53863	1.594154	0.0210838
810.61479	1.590757	0.0212489
658.55749	2.189021	0.0214305
1272.9818	Inf	0.0232296
824.6636	1.663926	0.0247375
610.55919	14.85244	0.0255087
337.29633	Inf	0.0258986
730.55306	1.813783	0.0261185
793.5823	1.806134	0.0264557
820.63392	1.632964	0.0274367
575.52214	Inf	0.031613
603.55277	22.94141	0.0319
638.58902	5.931064	0.0388209
630.63404	1.605405	0.0414139
612.57405	3.930728	0.0492175

Candidamtes from the *P. vivax*-infected samples in negative mode.

mz	Pv.fc	Pv.pval
459.3314	1.543587	0.0014423
761.5805	Inf	0.0239964
729.1757	2.549891	0.0294128
580.3583	Inf	0.0382051
745.6188	1.785265	0.0415542
888.5707	Inf	1.06E-05
690.5071	Inf	1.13E-05
828.575	2.919996	6.13E-05
842.5908	1.859495	0.0001387
840.5775	1.739445	0.0001687
824.578	1.904128	0.0001697
892.6052	8.167646	0.0002137
714.5086	11.76899	0.0004215
798.5269	4.264878	0.0006933
740.5215	2.969355	0.0007282
772.5137	37.85927	0.0007572
885.5488	2.247316	0.0007889
742.5383	2.671129	0.0008188

mz	Pv.fc	Pv.pval
830.59166	Inf	0.0008712
883.53157	Inf	0.000898
861.54857	Inf	0.000971
769.50064	Inf	0.0010684
864.5752	1.587473	0.0012232
800.5435	3.303532	0.0013314
844.60838	2.878574	0.0019292
859.53166	Inf	0.0021093
786.51475	2.135026	0.0022959
816.57691	2.544988	0.0023674
796.51398	2.684663	0.0028132
675.52039	Inf	0.003112
661.50383	Inf	0.0036286
833.51951	Inf	0.0040661
857.52068	Inf	0.004611
626.57122	5.627245	0.0046794
820.60644	1.918995	0.0048311
792.57732	1.62428	0.0056547

mz	Pv.fc	Pv.pval
738.50877	2.03754	0.0060964
881.5185	Inf	0.0063052
846.52767	1.71236	0.0063317
802.55867	1.74048	0.0092929
677.5351	Inf	0.0149494
680.61822	1.85811	0.0161746
637.50415	Inf	0.0278532
651.52134	36.8541	0.0288921
679.55165	9.14184	0.0379009

Candidates from the non-infected samples in positive mode.

mz	Pv.fc	Pv.pval
441.2991	0.637417	0.00491
473.4009	0.614827	0.00575
497.3867	0.317132	0.01322
552.1975	0.243012	0.01862
658.5275	0.302967	0.02061
699.5528	0.327823	0.03549
490.4276	0.625457	0.04142
536.1684	0.324028	0.04629
840.5984	0.312731	3.22E-06
842.6392	0.589598	5.46E-05
1347.887	0.455964	9.50E-05
837.6919	0.516885	0.00019
826.58	0.653843	0.00022
551.3475	0.65948	0.00025
750.5577	0.663863	0.0003
748.542	0.658015	0.00033
836.5566	0.619758	0.00035
1114.335	0.439974	0.0005

mz	Pv.fc	Pv.pval
774.55685	0.572997	0.000506
851.60118	0.327815	0.000711
857.49438	0.660142	0.001482
827.60191	0.425094	0.001515
495.286	0.645733	0.002073
427.2728	0.564253	0.002487
1024.6758	0.606021	0.002721
604.3876	0.57139	0.002941
647.47666	0.614239	0.003106
1300.836	0.287384	0.00364
1007.7165	0.641534	0.005316
780.71974	0.618856	0.008014
621.32951	0.653772	0.008236
638.35561	0.64362	0.008434
823.62004	0.535867	0.00887
714.48312	0.529103	0.01052
633.55949	0.533537	0.011686
683.64562	0.377469	0.019701

mz	Pv.fc	Pv.pval
827.7204	0.32344	0.026195
792.7201	0.62757	0.029838
737.4978	0.63098	0.035672
766.705	0.49791	0.039511
738.6743	0.51334	0.043884
891.2865	0.56262	0.046324
712.66	0.48682	0.047267

Candidates from the non-infected samples in negative mode.

mz	Pv.fc	Pv.pval
439.2819	0.4217546	0.010857
503.4061	0.6116275	0.018234
529.4272	0.3929495	0.02593
836.5783	0.2932807	1.06E-06
722.5115	0.4790224	1.72E-06
806.533	0.5173233	2.44E-06
850.5578	0.5835112	3.47E-06
869.6711	0.6483633	5.57E-06
1042.244	0.5386863	6.41E-06
812.539	0.4826483	7.24E-06
1015.222	0.5028831	7.65E-06
836.5442	0.466677	1.15E-05
774.5429	0.5565329	1.43E-05

mz	Pv.fc	Pv.pval
834.52942	0.44686	1.83E-05
857.67332	0.65852	1.95E-05
897.7048	0.48532	2.38E-05
810.52976	0.60267	3.48E-05
824.54417	0.58044	4.39E-05
1091.2397	0.6042	4.54E-05
746.16647	0.59397	5.30E-05
838.55974	0.25394	7.32E-05
1117.2619	0.59254	8.92E-05
871.69091	0.59585	0.0002726
719.14727	0.38191	0.0004649
1397.8869	0.63413	0.0005182
1165.2584	0.64362	0.0006727

mz	Pv.fc	Pv.pval
804.57512	0.661596	0.004305
403.30657	0.638247	0.005659
1192.2786	0.642934	0.010588
1266.2976	0.659079	0.017918
1340.3165	0.625666	0.044487
669.56719	0.655772	0.046189
965.24792	0.5065	0.047256